Neural Audio Effect Modeling
An Introduction

Hsiao Wen Yi 蕭文逸

Independent Research

Overview

- Introduction: Audio Effect
- Chapter I: Related Work
- Chapter II: HyperGRU for Neural AFx Modeling (Proposed Model)
- Chapter III: Future Work

- Formulation

$$y = f(x, c_t, c_g)$$

x: input signal, M channel
y: output signal, N channel
c_g: gloabl condition
c_t: local condition

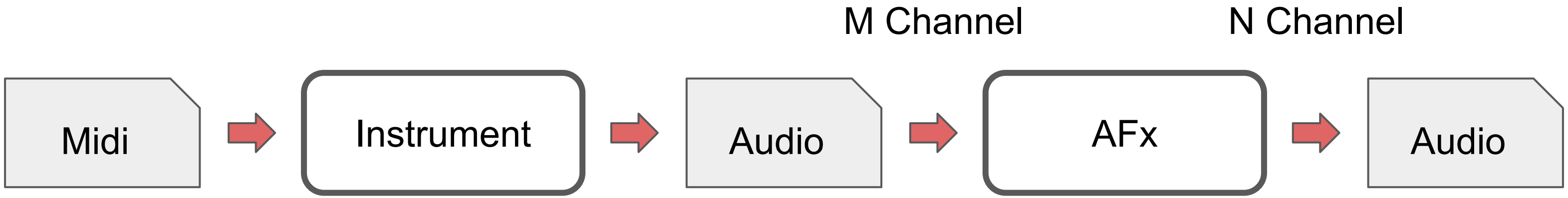
- Why Audio Effect Modeling
  - Analog Emulation
    - condition: knob values
  - Spatial/Immersive Audio (Virtual Reality)
    - condition: coordinates
- Why Neural Network
  - Quality
  - Differentiability: diverse application
  - Generalizability

00

Overview- Audio Effect

Introduction

Audio Effects



- Format: VSTi
- Application
  - Sampler
  - Sample library
  - Synthesizer
  - Wavetable
  - ...

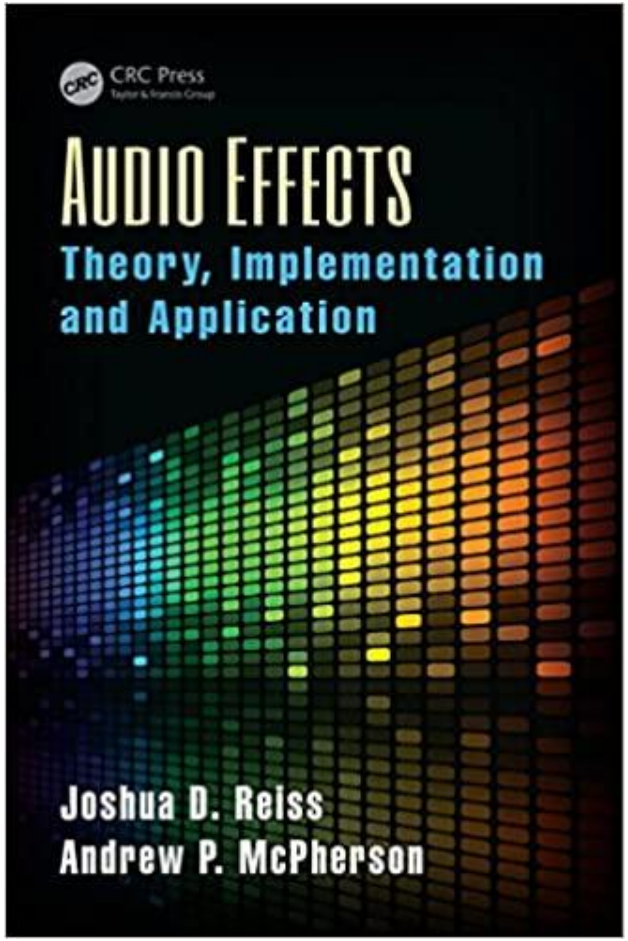
- Format: VST
- Application
  - Equalizer (EQ)
  - Distortion
  - Reverberation
  - Compressor/Limiter
  - …

$$y = f(x, c_t, c_g)$$

x: input signal, M channel
y: output signal, N channel
c_g: gloabl condition
c_t: local condition

Audio Effects

- Github: [juandagilc/Audio-Effects]
- Common audio effects list
  - EQ - Parametric EQ, Graphic EQ…
  - Dynamics - Compressor, Limiter, Expander, De-esser
  - Distortion - Overdrive pedal, Amp, Saturator
  - Reverb - Chamber, Hall, Room, Plate…
  - Delay - Spring delay, Tape delay, Ping-pong delay…
  - Modulation - Flanger, Chorus, Phaser
  - Spatial - Stereo imager, Mid/Side processor
  - Others - Noise reduction, Pitch-correction…



01

Related Work

Chapter I

Chapter 1 - Related Work

- Audio Effect Modeling
  - Traditional DSP
  - Neural Networks
  - DDSP
- Condition in Neural Networks
  - Concatenation
  - FiLM
  - HyperNetworks
- Intrinsic Problem of Neural Networks
  - Aliasing
  - Chaos

Chapter 1 - Related Work

- **Audio Effect Modeling**
  - **Traditional DSP**
  - **Neural Networks**
  - **DDSP**
- Condition in Neural Networks
  - Concatenation
  - FiLM
  - HyperNetworks
- Intrinsic Problem of Neural Networks
  - Aliasing
  - Chaos

Traditional DSP

- Impulse Response (IR)
- White-Box
  - Characteristic function
  - Circuit analysis
- Black-Box
  - Wiener-Hammerstein (WH) models
- Hybrid Method
  - Build a guitar amplifier
- Discussion

Traditional DSP: Impulse Response

- Assumption:
  - Linear Time-Invariant (LTI) System
- Application
  - Guitar cabinet
  - Room reverberation (RIR)
  - **H**ead **R**elated **T**ransfer **F**unctions (HRTF)
- Convolution
- Pros
  - Fast and simple
- Cons
  - Non-linear, time-variant, memory

Traditional DSP: Characteristic Curve

- White-Box
- Signal Clipping
- Waveshaping

Traditional DSP: Circuit Analysis

- White-Box
- Nodal Analysis
  - Rewrite the schematic into equations
- pros:
  - Accurate
  - User control
- cons:
  - Slow and infeasible for large circuit
  - Re-design everytime
  - Need to open up the hardware
  - not for all modules

Example: Guitar Tone Stack

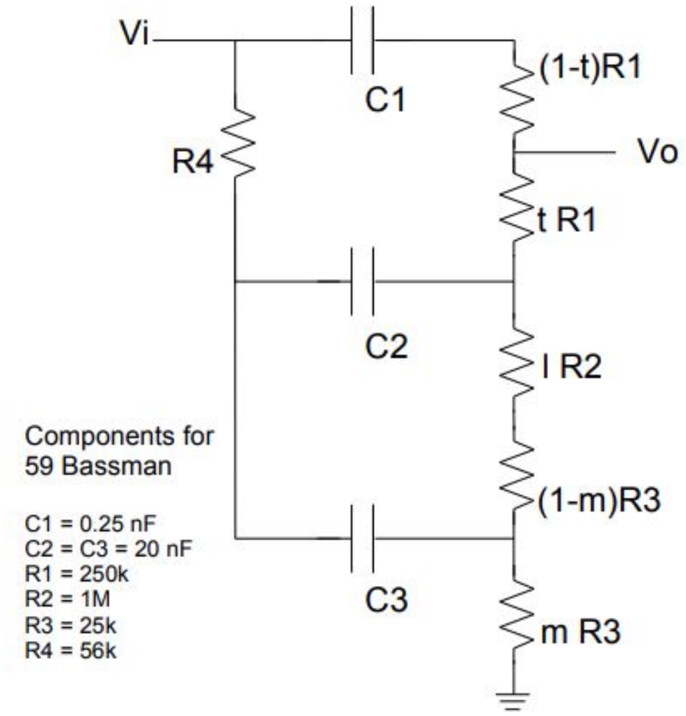


Figure 1: *Tone stack circuit with component values.*

(DAfx'06) DISCRETIZATION OF THE '59 FENDER BASSMAN TONE STACK, David T. Yeh

Traditional DSP: Circuit Analysis

- *How Waves' Modeling Captures Analog Magic in a Digital World* from **Waves' Blog**
  - "The first step in this kind of modeling is to open up the hardware..."
  - component by component
  - If there are too many components, simplification is necessary.
  - "write mathematical equations that quantify how the components perform" in MATLAB
  - "the modeling process takes months—in extreme cases even years—..."
- Expensive

$$
\begin{aligned}
b_3 &= lm(C_1C_2C_3R_1R_2R_3 + C_1C_2C_3R_2R_3R_4) \\
&\quad - m^2(C_1C_2C_3R_1R_3^2 + C_1C_2C_3R_3^2R_4) \\
&\quad + m(C_1C_2C_3R_1R_3^2 + C_1C_2C_3R_3^2R_4) \\
&\quad + tC_1C_2C_3R_1R_3R_4 - tmC_1C_2C_3R_1R_3R_4 \\
&\quad + tlC_1C_2C_3R_1R_2R_4,
\end{aligned}
$$

$$
\begin{aligned}
a_0 &= 1, \\
a_1 &= (C_1R_1 + C_1R_3 + C_2R_3 + C_2R_4 + C_3R_4) \\
&\quad + mC_3R_3 + l(C_1R_2 + C_2R_2), \\
a_2 &= m(C_1C_3R_1R_3 - C_2C_3R_3R_4 + C_1C_3R_3^2 \\
&\quad + C_2C_3R_3^2) + lm(C_1C_3R_2R_3 + C_2C_3R_2R_3) \\
&\quad - m^2(C_1C_3R_3^2 + C_2C_3R_3^2) + l(C_1C_2R_2R_4 \\
&\quad + C_1C_2R_1R_2 + C_1C_3R_2R_4 + C_2C_3R_2R_4) \\
&\quad + (C_1C_2R_1R_4 + C_1C_3R_1R_4 + C_1C_2R_3R_4 \\
&\quad + C_1C_2R_1R_3 + C_1C_3R_3R_4 + C_2C_3R_3R_4), \\
a_3 &= lm(C_1C_2C_3R_1R_2R_3 + C_1C_2C_3R_2R_3R_4) \\
&\quad - m^2(C_1C_2C_3R_1R_3^2 + C_1C_2C_3R_3^2R_4) \\
&\quad + m(C_1C_2C_3R_3^2R_4 + C_1C_2C_3R_1R_3^2 \\
&\quad - C_1C_2C_3R_1R_3R_4) + lC_1C_2C_3R_1R_2R_4 \\
&\quad + C_1C_2C_3R_1R_3R_4,
\end{aligned}
$$

(DAfx'06) DISCRETIZATION OF THE '59 FENDER BASSMAN TONE STACK, David T. Yeh

Traditional DSP: WH model

- Black-Box
- Wiener-Hammerstein (WH) model
  - Linear -> Non-linear -> Linear
- Loss Optimization
  - Levenberg–Marquardt method (gradient-based)
- Pros
  - avoid exhaustive anaylsis
- Cons
  - Configuration
  - Performance
  - No user control

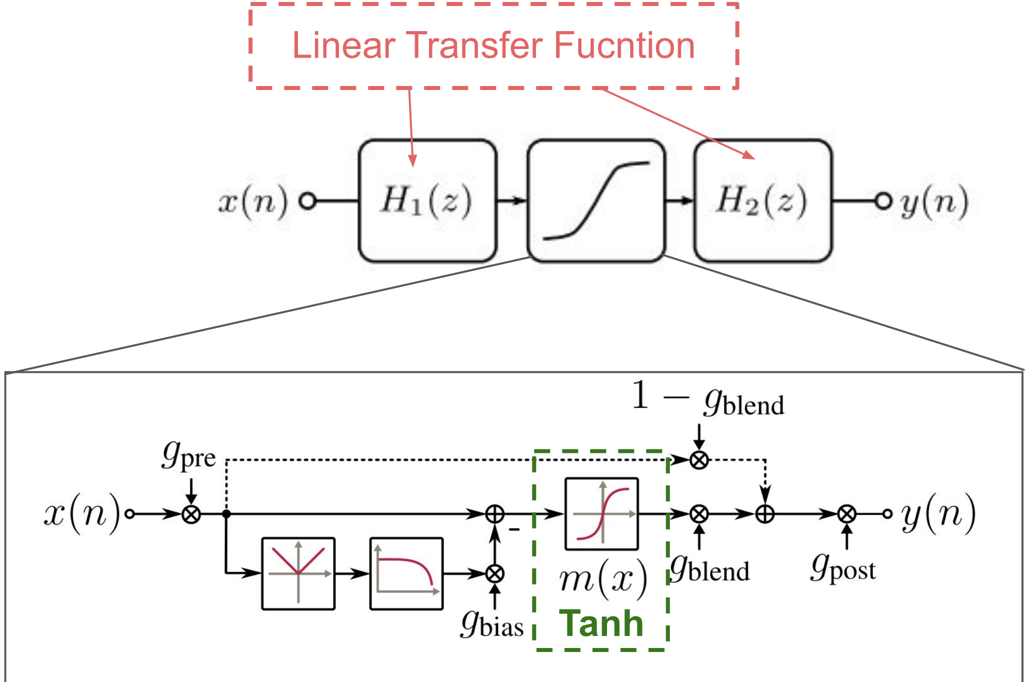
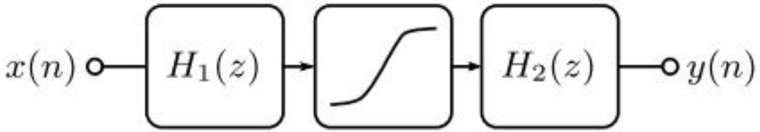


Figure 5: Signal flow graph of the used nonlinear block.

(DAGA'18) *Virtual Analog Modeling of Guitar Amplifiers with Wiener-Hammerstein Models, Felix Eichas*

Traditional DSP: WH model



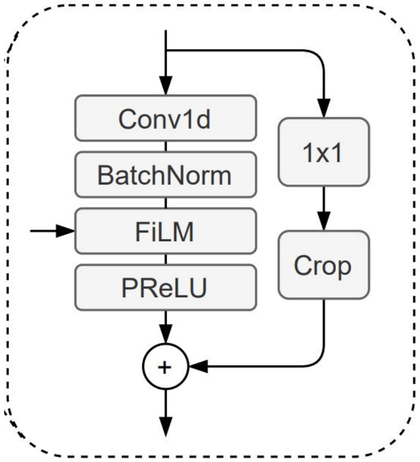
WH Model

- Wiener-Hammerstein (WH) model
  - Linear -> Non-linear -> Linear
  - Gradient based optimization

- Similarity with Modern Neural Networks

> guitar distortion effects. The TCN is a generalization of convolutional networks applied to sequence modeling (dilated 1-dimensional convolution + nonlinearity). Interestingly, yet maybe somewhat unsurprisingly, these models resemble Wiener-Hammerstein models [26], a traditional statistical approach to

From micro-tcn v1 paper

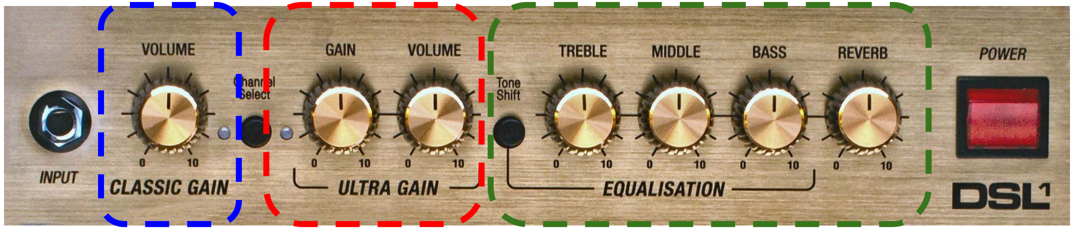


Micro-TCN Block

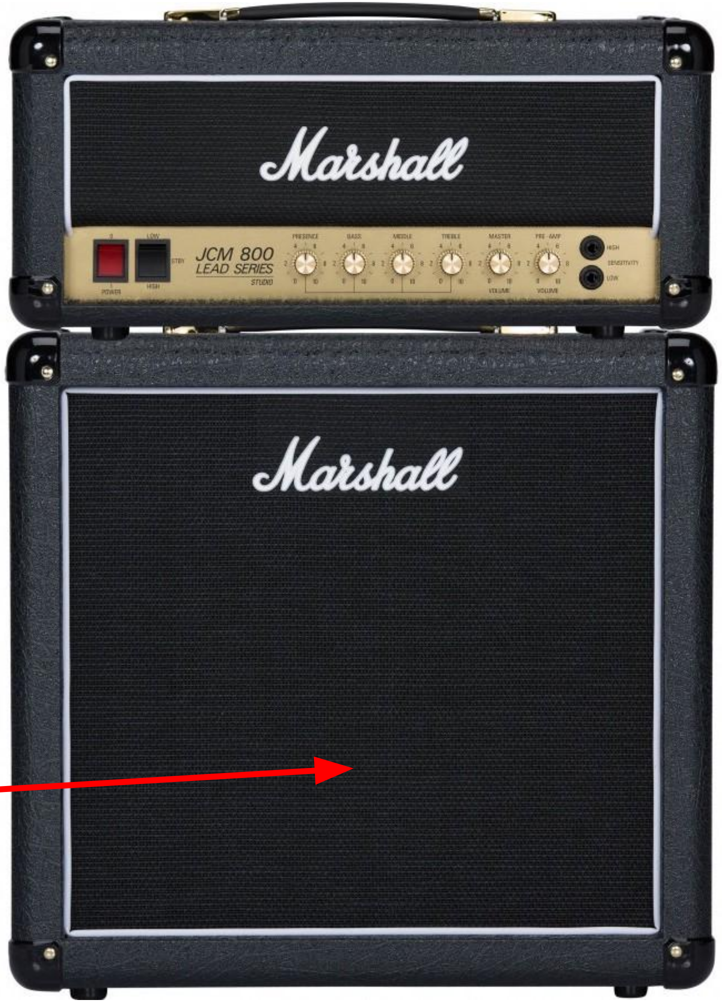
(AES'22) Efficient neural networks for real-time modeling of analog dynamic range compression, Christian J.

Traditional DSP: Hybrid Method

- How to build a **guitar amplifier**?

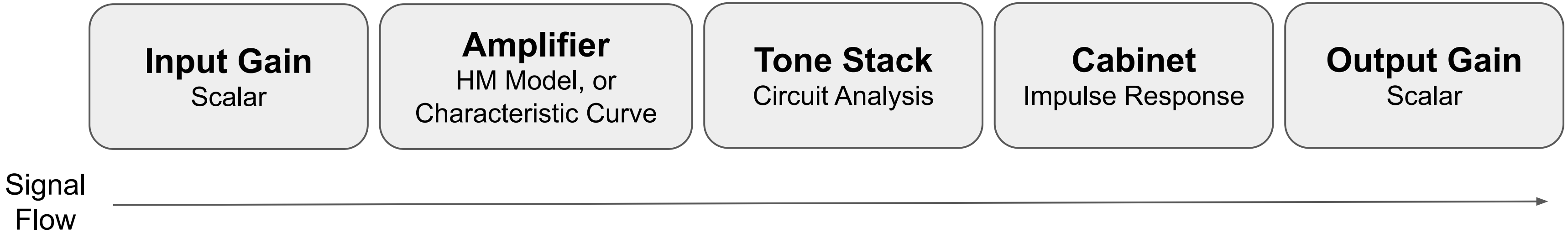


- Input Gains: Degree of distortion
- Tone Stack: Equalization
- Output Gain: Volume
- Cabinet



Traditional DSP: Hybrid Method

- How to build a **guitar amplifier**?



Traditional DSP: Discussion

- Other Methods
  - Voletrra Serires [1] (Adopted by Acustica Audio)
  - Wave Digital Filter (WDF) [2]
- Problems
  - Based on certian assumptions, lack of generalizability
  - Some methods are resource demanding and slow
  - Manual analysis and handcrafted features are usually required
  - Quality

[1] (JAES'18) Identification of volterra models of tube audio devices using multiple-variance method
[2] (Icassp'06) Wave digital simulation of a vacuumtube amplifier

Neural Networks

- Researcher
  - Marco A. Martinez Ramirez
  - Christian J. Steinmetz
  - **Vesa Välimäki**
    - Professor@Aalto University
  - **Alexander Richard**
    - Research Scientist@Meta Reality Labs

- Architectures
  - TCNs
  - RNNs
  - Others

Researcher: Marco A. Martinez Ramirez

- Experience
  - PhD@QML
  - Intern@Adobe Research
  - Researcher@Sony
- Info
  - Website
  - Google Scholoar
  - Github

Research Areas

~deep learning architectures for music and audio processing.

~intelligent music production: automatic mixing and mastering.

~audio effects and neural networks.

~DSP-informed machine learning.

Researcher: Marco A. Martinez Ramirez

- (Dafx’18) End-to-end Equalization with Convolutional Neural Networks
- (Icassp’19) Modeling Nonlinear Audio Effects with End-to-end Deep Neural Networks
- (Dafx’19) A General-Purpose Deep Learning Approach to Model Time-Varying Audio Effects
- (ApplSci’20) Deep Learning for Black-Box Modeling of Audio Effects
- (Icassp’20) Modeling Plate and Spring Reverberation Using A DSP-Informed Deep Neural Network

DSP-informed UNet

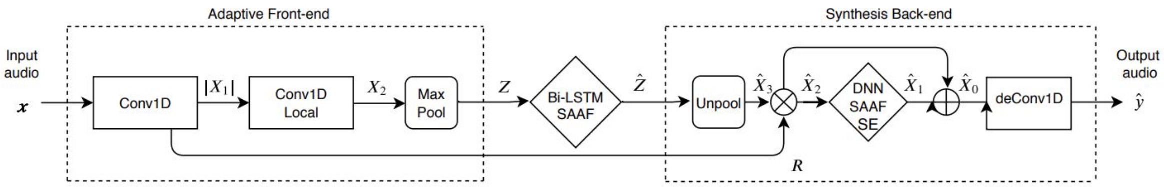


Figure 1: *Block diagram of the proposed model; adaptive front-end, Bi-LSTM and synthesis back-end.*

- (Icassp’21) Differentiable Signal Processing With Black-Box Audio Effects
- (Icassp’22) Automatic DJ Transitions with Differentiable Audio Effects and Generative Adversarial Networks
- (arXiv.2202) Removing Distortion Effects in Music Using Deep Neural Networks

Researcher: Christian J. Steinmetz

- Experience
  - PhD@QML
  - Intern@Adobe Research
- Info
  - Website
  - Google Scholoar
  - Github

about

I am a PhD student working with Prof. Joshua D. Reiss within the Centre for Digital Music at Queen Mary University of London. I research applications of machine learning in audio with a focus on differentiable signal processing. Currently, my research revolves around high fidelity audio and music production, which involves enhancing audio, intelligent systems for audio engineering, as well as applications of machine learning that augment and extend creativity.

Previously, I was an intern at Adobe, Meta AI, Dolby, Bose, Tape It, and Cirrus Logic.

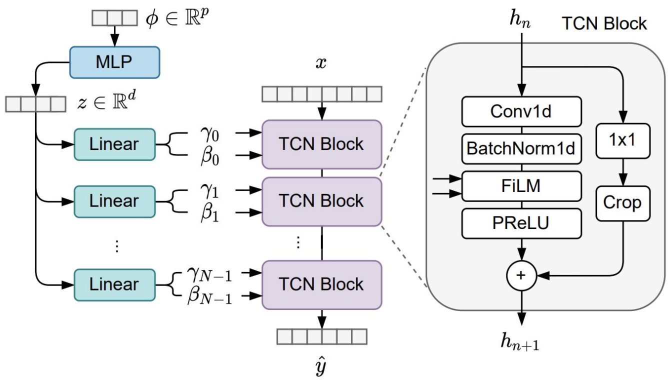
GitHub • Scholar • Twitter • YouTube

Researcher: Christian J. Steinmetz

- (arXiv.2010) Randomized Overdrive Neural Networks
- (DMRN+15) auraloss: Audio-Focused Loss Functions in PyTorch
- (Aes'21) pyloudnorm: A Simple yet Flexible Loudness Meter in Python
- (Icassp'21) Automatic Multitrack Mixing With A Differentiable Mixing Console Of Neural Audio Effects
- (NeurIPS'21) Steerable Discovery of Neural Audio Effects (ML4CD Workshop)
- (Aes'22) Efficient Neural Networks for Real-Time Modeling of Analog Dynamic Range Compression
- (Icassp'22) Direct Design of Biquad Filter Cascades with Deep Learning by Sampling Random Polynomials

TCN

Auto Mixing

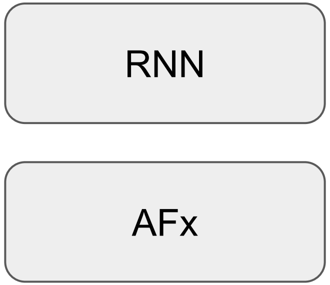


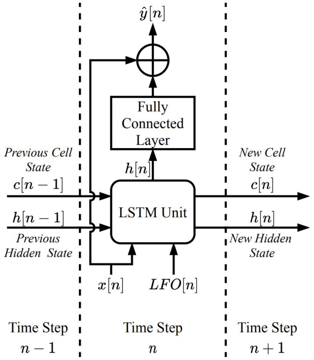
Researcher: Vesa Välimäki

- Experience
  - Professor@Aalto University
- Info
  - Website
  - Google Scholoar (~12000 citations)
- Industry
  - Several alumni working at Nueral DSP

Researcher: Vesa Välimäki

- (SMC’19) Real-Time Modeling of Audio Distortion Circuits with Deep Learning
- (Icassp’19) Deep Learning for Tube Amplifier Emulation
- (Dafx’19) Real-Time Black-Box Modelling With Recurrent Neural Networks
- (ApplSci’20) Real-Time Guitar Amplifier Emulation with Deep Learning
- (Icassp’20) Perceptual Loss Function for Neural Modeling of Audio System
- (Dafx’20) Neural Modelling of Periodically Modulated Time-Varying Effects
- (Dafx’21) Exposure bias and state matching in recurrent neural network virtual analog models
- (Dafx’22) Virtual Analog Modeling of Distortion Circuits Using Neural Ordinary Differential Equations





Researcher: Alexander Richard

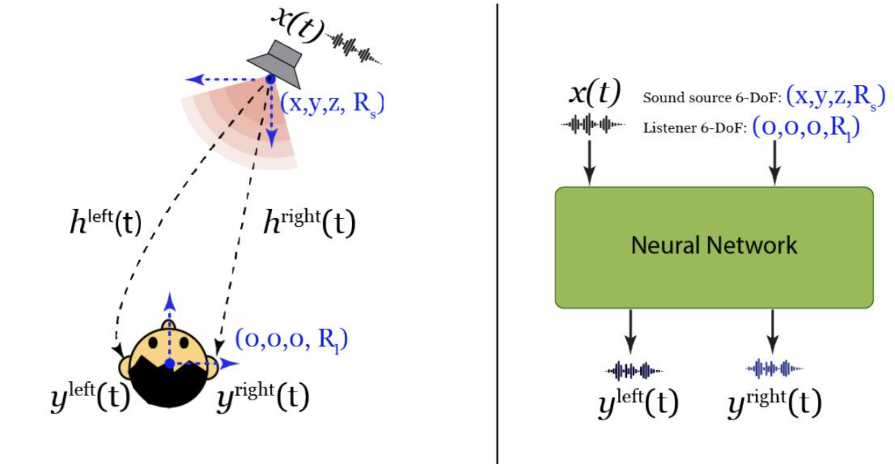
- Experience
  - Research scientist@Meta Reality Labs
- Info
  - [Website]
  - [Google Scholoar]

Researcher: Alexander Richard

- (Icassp'21) Implicit Hrtf Modeling Using Temporal Convolutional Networks
- (ICLR'21) Neural Synthesis of Binaural Speech from Mono Audio
- (Icassp'22) Deep Impulse Responses: Estimating and Parameterizing Filters with Deep Networks

TCN

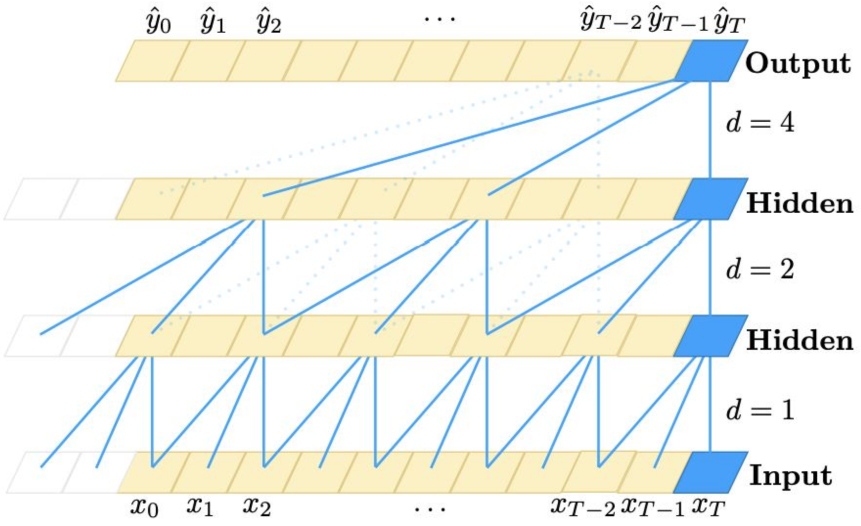
Binaural



(a) Traditional synthesis system (b) Proposed system

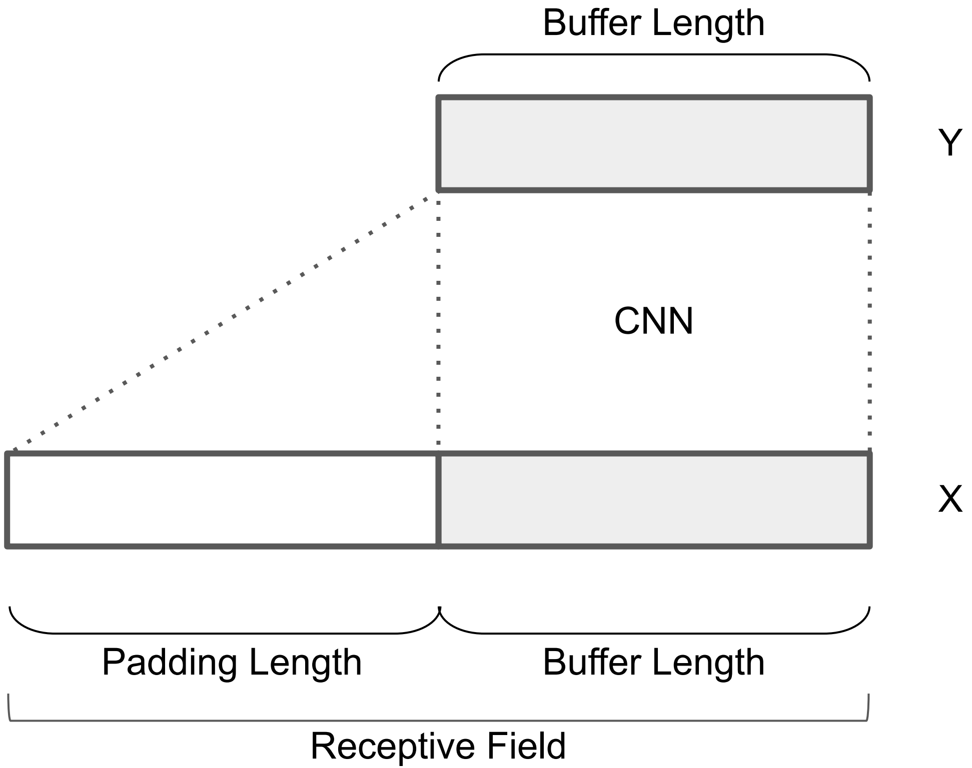
Architectures: TCNs

- **T**emporal **C**onvolutional **N**etworks
  - An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (arXiv.1803)
- **TCN = 1D Causal Dilated Convolutions**
- Family (with proper **modification**)
  - Wavenet
  - TCN
  - Micro-TCN
- Difference:
  - Activation
  - Residual design
  - Kernel design



Architectures: TCNs

- Modification
  - Causality
  - Padding Policy
    - zeros
    - cached samples



Architectures: RNNs

Vesa Välimäki

WaveNet

- (SMC'19) Real-Time Modeling of Audio Distortion Circuits with Deep Learning
- (Icassp'19) Deep Learning for Tube Amplifier Emulation

RNN

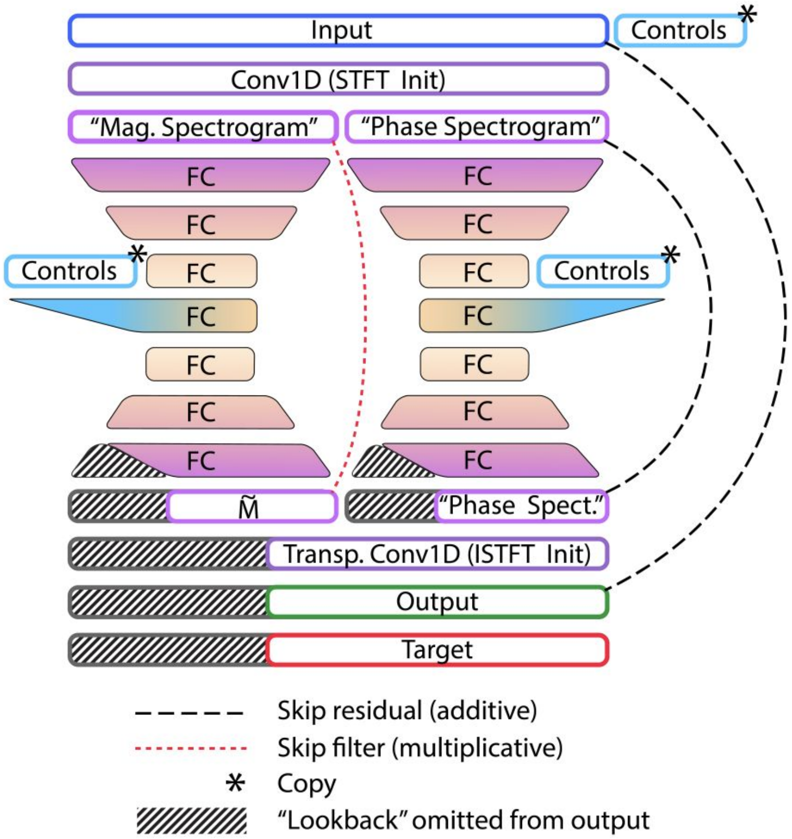
- **(Dafx'19) Real-Time Black-Box Modelling With Recurrent Neural Networks**
- **(ApplSci'20) Real-Time Guitar Amplifier Emulation with Deep Learning**
- **(Icassp'20) Perceptual Loss Function for Neural Modeling of Audio System**
- **(Dafx'20) Neural Modelling of Periodically Modulated Time-Varying Effects**
- **(Dafx'21) Exposure bias and state matching in recurrent neural network virtual analog models**

Neural ODE

- (Dafx'22) Virtual Analog Modeling of Distortion Circuits Using Neural Ordinary Differential Equations

Architectures: Others

- UNet
  - Marco A. Martinez Ramirez
  - [SignalTrain]
- Not Good :(



DDSP

- Differentiable IIR
- Differentiable Circuit
- Others

DDSP: Differentiable IIR

- (Dafx’20) Neural Parametric Equalizer Matching Using Differentiable Biquads
- (Dafx’20) Differentiable IIR filters for machine learning applications
- (Dafx’20) Optimization of cascaded parametric peak and shelving filters with backpropagation algorithm
- (Icassp’21) Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads
- (Icassp’22) Direct design of biquad filter cascades with deep learning by sampling random polynomials

Model	Params.	MSE
Coefficient	274	0.1708
Pole/zero	274	0.0885
Param. EQ	**210**	**0.0629**
WaveNet	22960	**0.0088**

Table 2. Model comparisons.

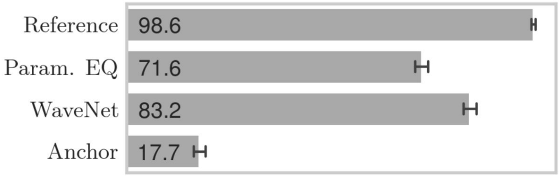


Fig. 3. MUSHRA scores with 95% confidence intervals.

DDSP: Differentiable Circuit

- (Dafx'20) Differentiable White-Box Virtual Analog Modeling

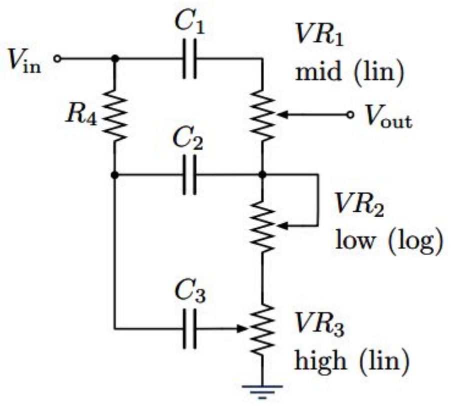


Figure 4: Schematic for the FMV Tone Stack.

	Value		
Name (λ)	Initial	Learned	G_λ
VR_1	250 kΩ	312 kΩ	1.2498
VR_2	1 MΩ	616 kΩ	0.6164
VR_3	25 kΩ	32 kΩ	1.2836
R_4	56 kΩ	29 kΩ	0.9081
C_1	250 pF	327.5 pF	1.3102
C_2	20 nF	17.3 nF	0.8652
$C3$	20 nF	16.8 nF	0.8408
w_1	0.566	0.5036	–
w_2	4.400	4.2547	–
b_1	−3.380	−3.3351	–
b_2	0.564	0.5016	–

Table 1: Intial and learned values for the FMV Tone Stack Model.

DDSP: Others

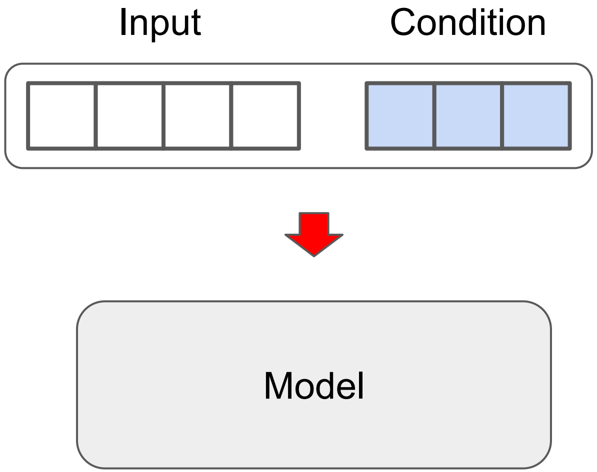
- DTW (Dynamic Time Warping)
  - (Iclr’21) Neural Synthesis of Binaural Speech From Mono Audio
- Reverberation
  - (arXiv.2105) Differentiable Artificial Reverberation

Chapter 1 - Related Work

- Audio Effect Modeling
  - Traditional DSP
  - Neural Networks
  - DDSP
- **Condition in Neural Networks**
  - **Concatenation**
  - **FiLM**
  - **HyperNetworks**
- Intrinsic Problem of Neural Networks
  - Aliasing
  - Chaos

Concatenation

- Simplest
- Most Common



FiLM

- (AAAI'18) FiLM: Visual Reasoning with a General Conditioning Layer
- (NeurIPS'19) Temporal FiLM: Capturing Long-Range Sequence Dependencies with Feature-Wise Modulations

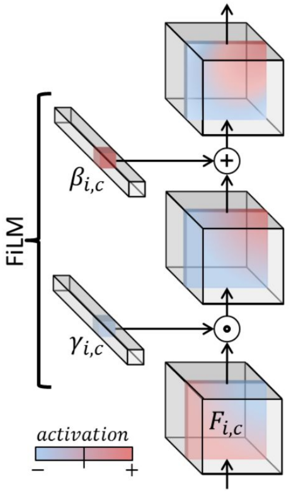


Figure 2: A single FiLM layer for a CNN. The dot signifies a Hadamard product. Various combinations of γ and β can modulate individual feature maps in a variety of ways.

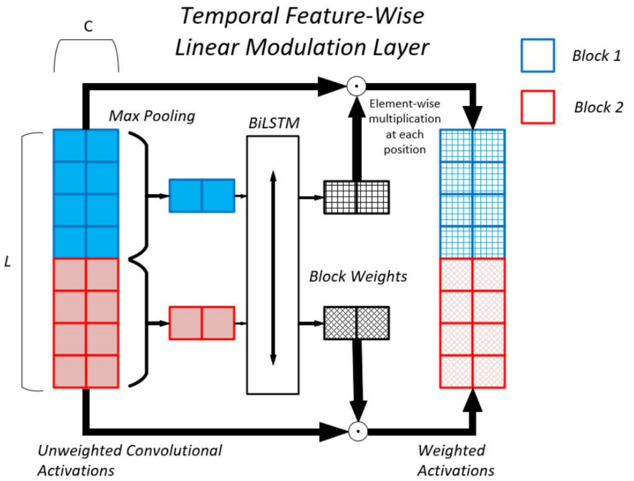


Figure 1: The TFiLM layer combines the strengths of convolutional and recurrent neural networks. *Above*: operation of the TFiLM layer with $T = 8$, $C = 2$, $B = 2$, and a pooling factor of 2.

FiLM: AFx

- (Icassp'21) Differentiable Mixing Console (DMC)
- (AES'22) micro-TCN

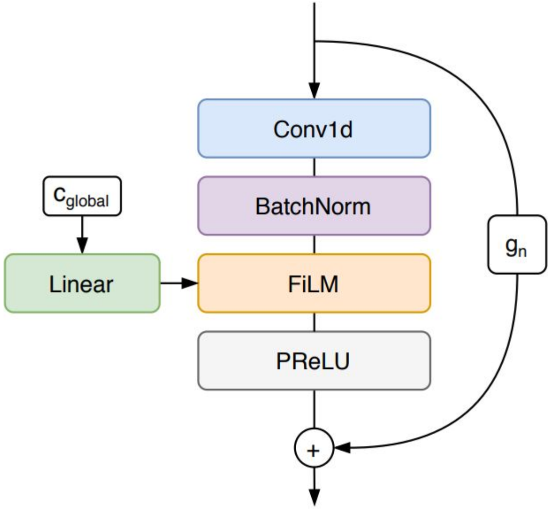


Fig. 2. Block diagram of the TCN block.

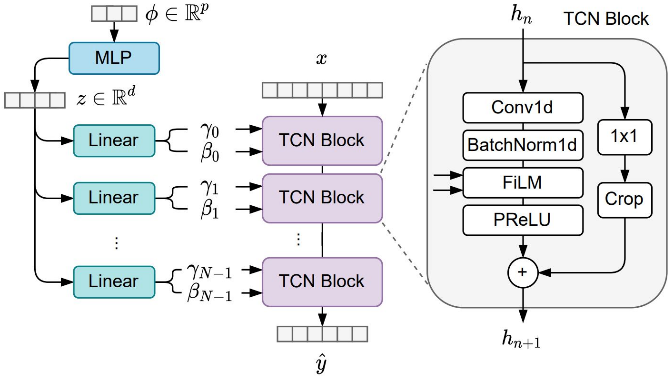


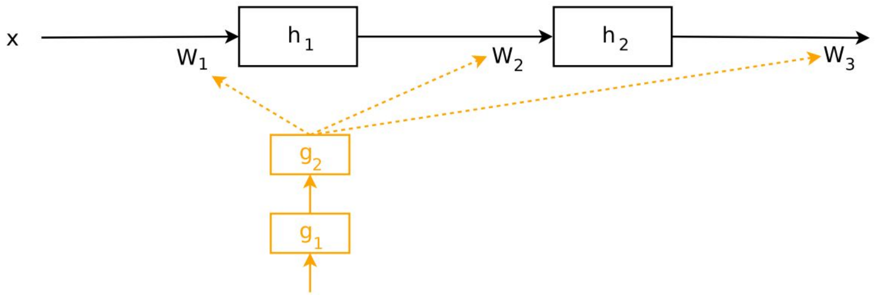
Fig. 1: TCN [20] with a series of convolutional blocks along with conditioning module (MLP) that adapts the gain γ_n and bias β_n at each layer as a function of the control parameters ϕ.

HyperNetworks

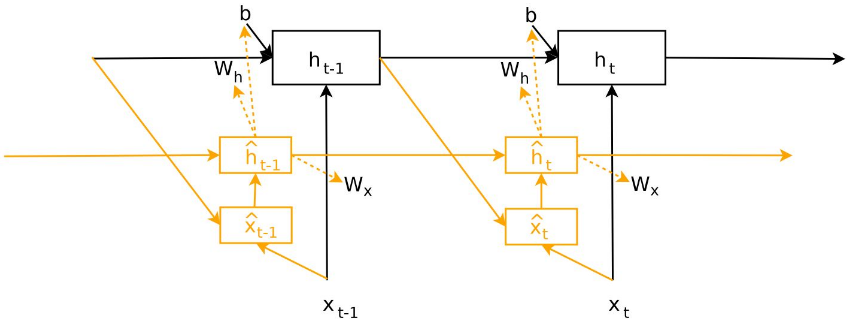
- (ICLR’17) HyperNetworks

  In this work, we consider an approach of using a small network (called a “hypernetwork") to generate the weights for a larger network (called a main network).

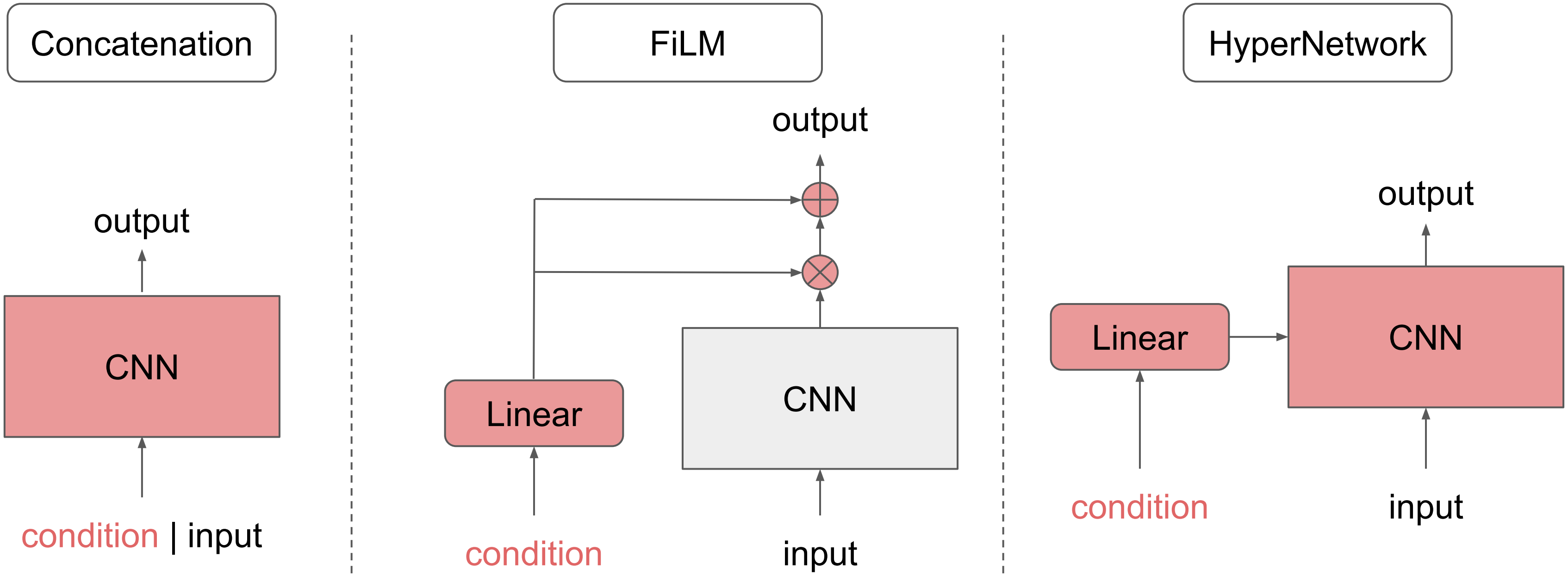
Static: Linear-CNN



Dynamic: RNN-RNN



HyperNetworks



HyperNetwork > FiLM : effectness of condition injection

HyperNetwork > Concatenation : efficiency in computation, *avoid DSP issues*

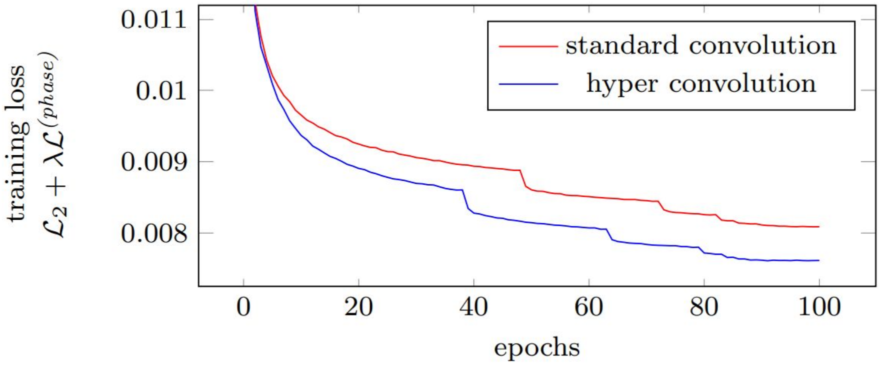
HyperNetworks: AFx

- (Icassp'21) [Implicit Hrtf Modeling Using Temporal Convolutional Networks]
- (ICLR'21) **[Neural Synthesis of Binaural Speech from Mono Audio]**

Model: HyperConv

Table 2: Ablation study. The components of the proposed binauralization network improve phas and amplitude and thereby the overall loss in time-domain.

		raw waveform (ℓ_2 error $\times 10^3$)	power spectrum (ℓ_2 error)	phase spectrum (angular error)
(a)	vanilla temporal CNN	0.254	0.061	0.934
(b)	+ warping	0.206	0.061	0.849
(c)	+ hyper-conv	0.183	0.051	0.847
(d)	+ sine activation	**0.167**	**0.048**	**0.807**



HyperNetworks: AFx

From Izotope

- (Icassp’21) Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads

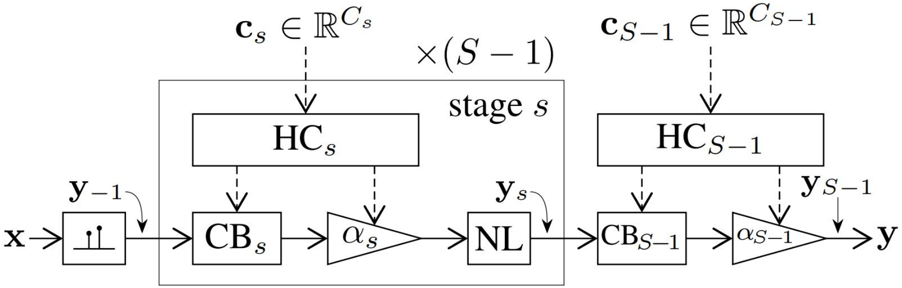


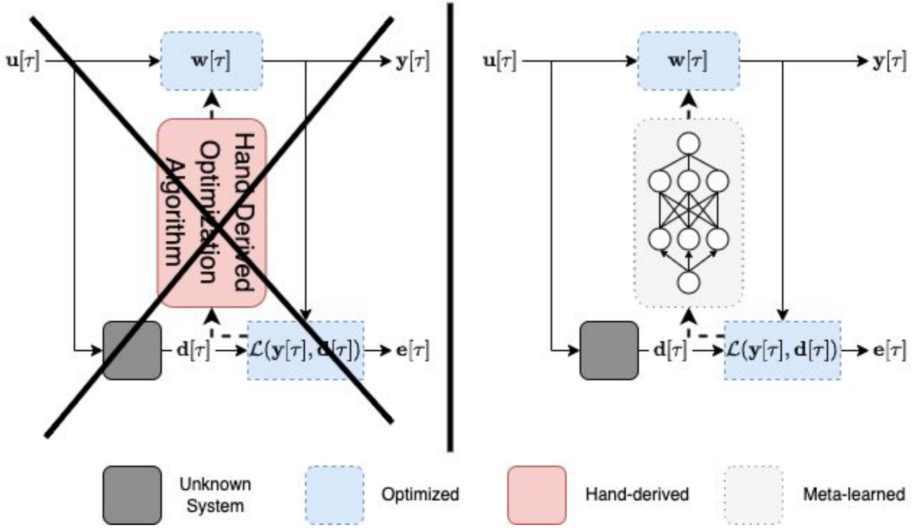
Fig. 2. Proposed neural network model architecture.

Architecture:
- HyperNet: MLP
- Main Net: IIR

HyperNetworks: DSP

From Adobe Research

- (WASPAA'21) Auto-DSP: Learning to Optimize Acoustic Echo Cancellers, Automatic Echo Cancellation
- (arXiv.2204) **Meta-AF: Meta-Learning for Adaptive Filters**



source: twitter

Architecture:

- HyperNet: RNN
- Main Net: Filters

Meta Learning: Learn new takss by self-supervision:

- system identification
- echo cancellation
- prediction
- dereveberation
- beamforming
- noise cancellation

Chapter 1 - Related Work

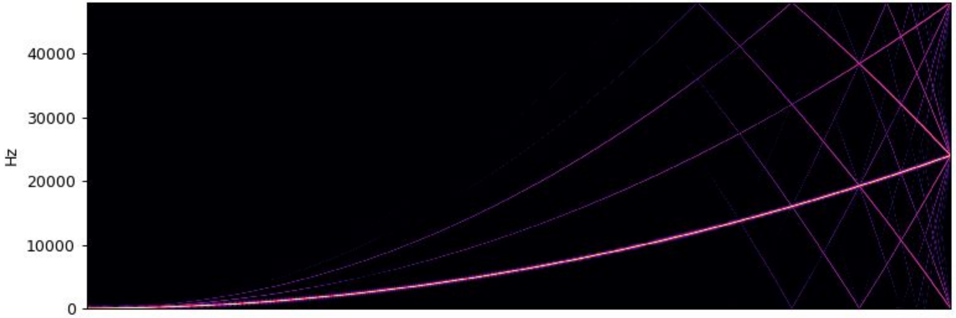
- Audio Effect Modeling
  - Traditional DSP
  - Neural Networks
  - DDSP
- Condition in Neural Networks
  - Concatenation
  - FiLM
  - HyperNetworks
- **Intrinsic Problem of Neural Networks**
  - **Aliasing**
  - **Chaos**

Aliasing

- Cause
  - Downsampling
  - Non-Linear function
- Solution
  - Oversampling + LPF
  - Anti-Aerivative
- Deep Learning

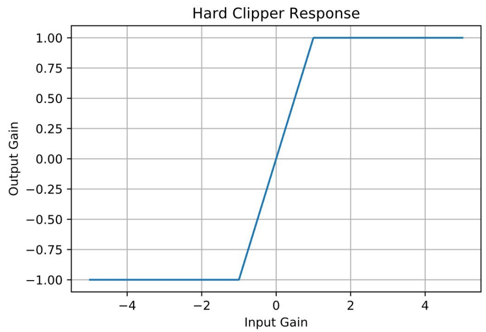
Image

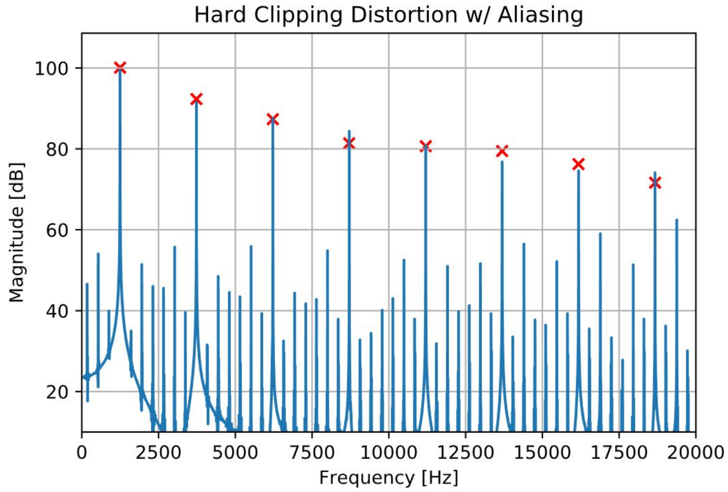
Audio

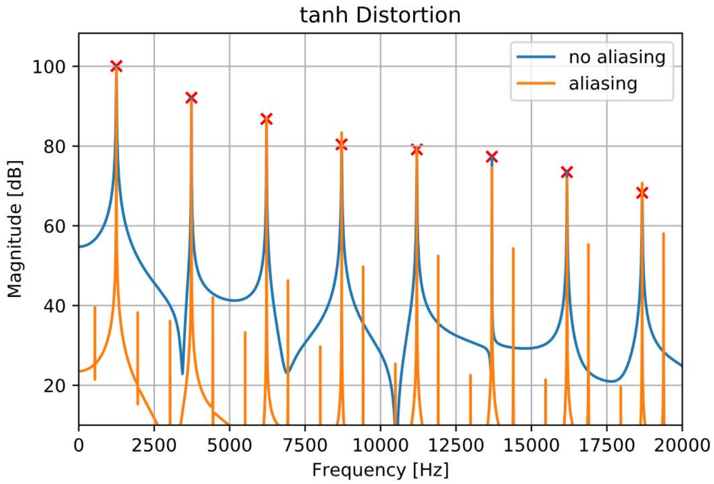


Aliasing

- Cause 1: Dowsampling
  - Sampling theorem: Nyquist Frequency
- Cause 2: None-Linear Function





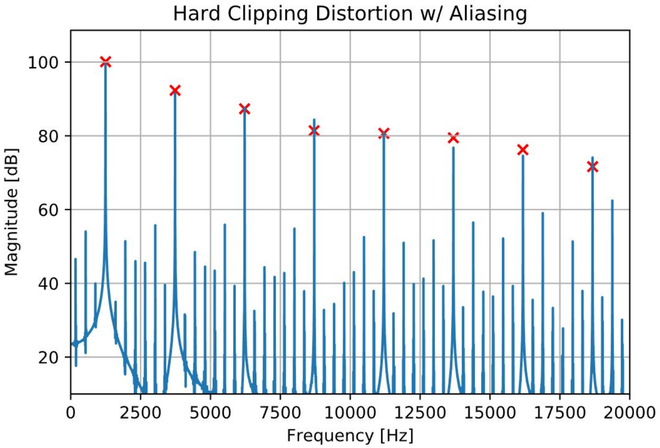


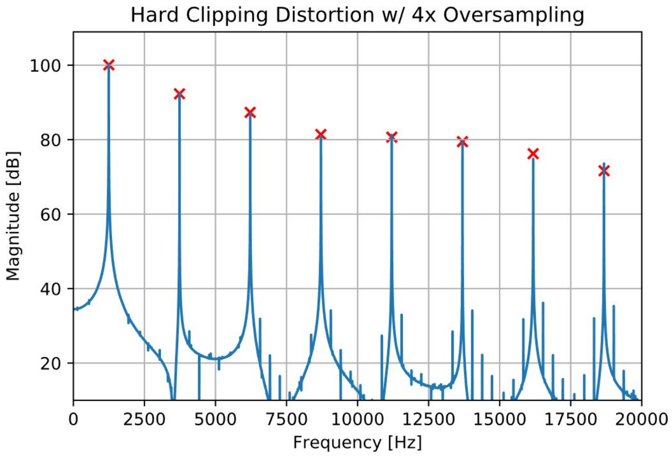
source: Jatin Chowdhury. [Github]() | [Medium]() | [Notebook]()

Aliasing

- Solution 1: Oversampling + Low-Pass Filter (LPF)

X → Upsampling → Non-Linear → LPF → Downsampling → Y

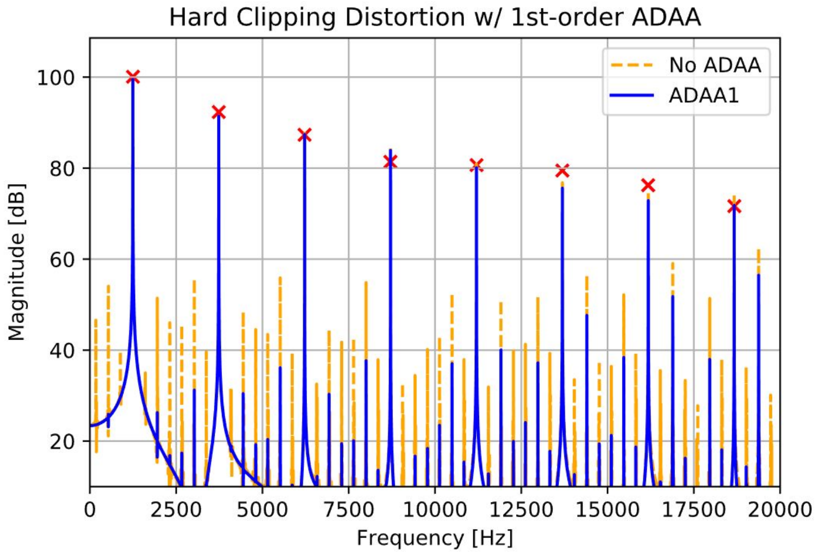




Aliasing

- Solution 2: Anti-Derivative Anti-Aliasing (ADAA)

(Dafx'16) Reducing the Aliasing of Nonlinear Waveshaping Using Continuous-Time Convolution
(SPL'17) Antiderivative Antialiasing for Memoryless Nonlinearities



Aliasing: Deep Learning

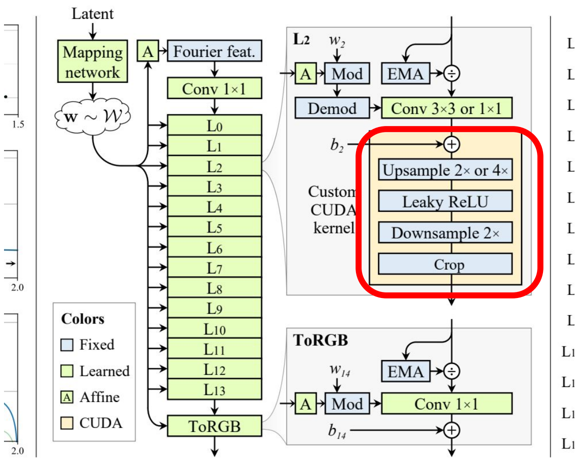
- Non-Linear Activation
- Oversampling

(ICML'19) Making Convolutional Networks Shift-Invariant Again
(NeurIPS'21) Alias-Free Generative Adversarial Networks (StyleGAN3)

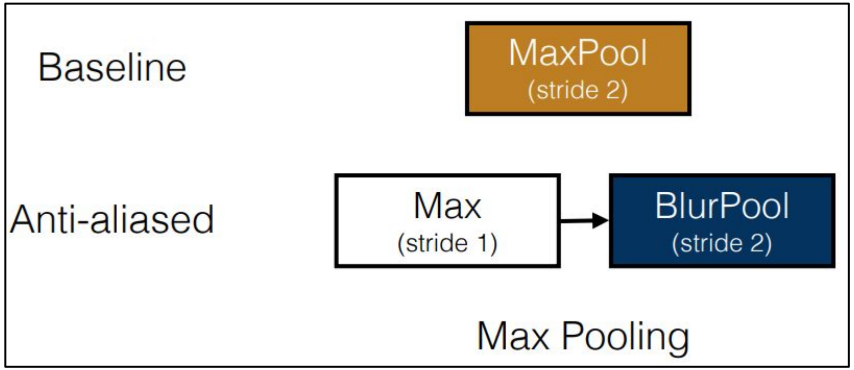
Low-Pass Filter

Blur Kernel

Sinc Filter



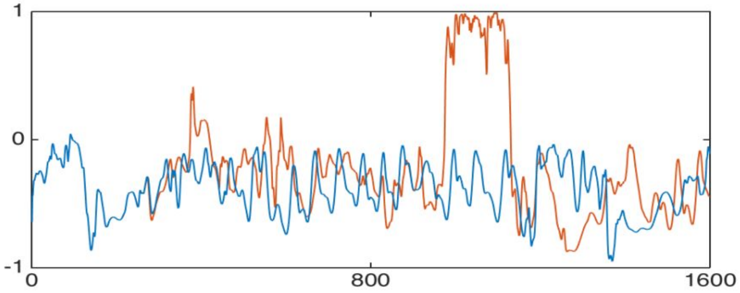
(b) Our alias-free StyleGAN3 generator architecture



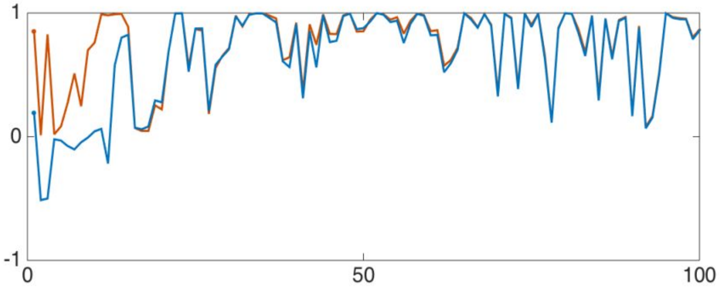
Chaos

- (ICLR'17) A Recurrent Neural Network without Chaos

When the input is absent, the trajectory of RNN states is not predictable



(a) No input data



(b) With input data

HyperGRU for Neural AFx Modeling

Chapter II

02

HyperGRU for Neural AFx Modeling

Chapter II

Chapter 2 - HyperGRU for Neural AFx Modeling

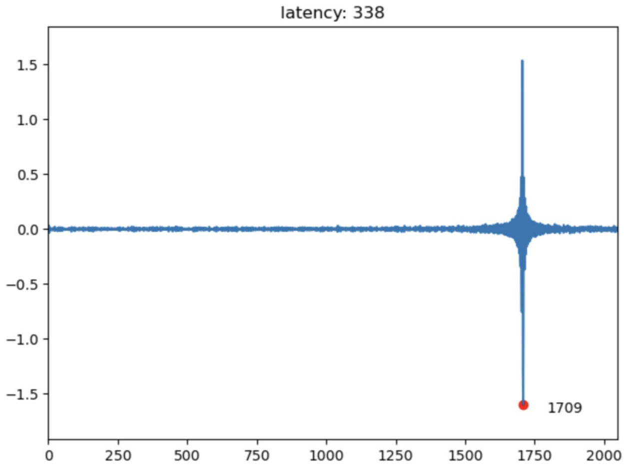
- Current Progress
  - Model
  - Dataset
  - Loss
  - Baselines
  - Experiments
  - Deployment
- The Palette
  - Tone Creation
  - Discussion
- Future Work
  - Advanced Model Design
  - Benchmark
  - Discussion

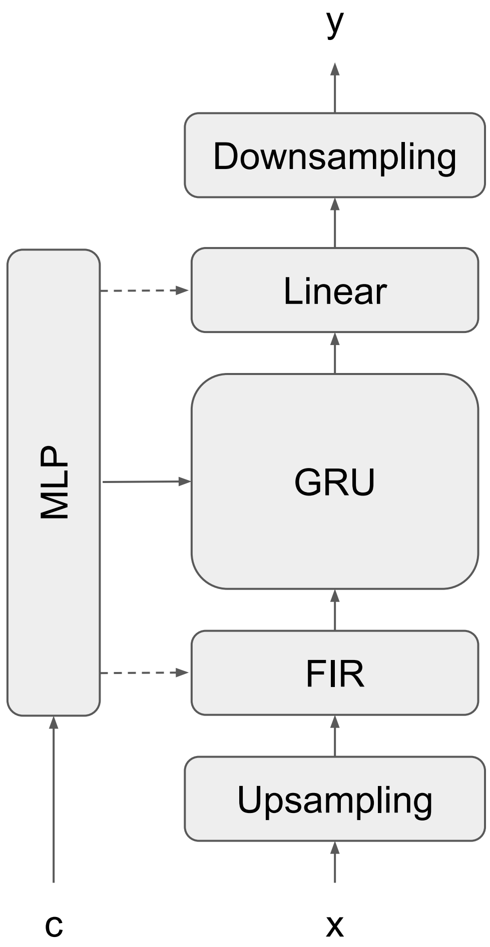
Chapter 2 - HyperGRU for Neural AFx Modeling

- **Current Progress**
  - **Model**
  - **Dataset**
  - **Loss**
  - **Baselines**
  - **Experiments**
  - **Deployment**
- The Palette
  - Tone Creation
  - Discussion
- Future Work
  - Advanced Model Design
  - Benchmark
  - Discussion

Model

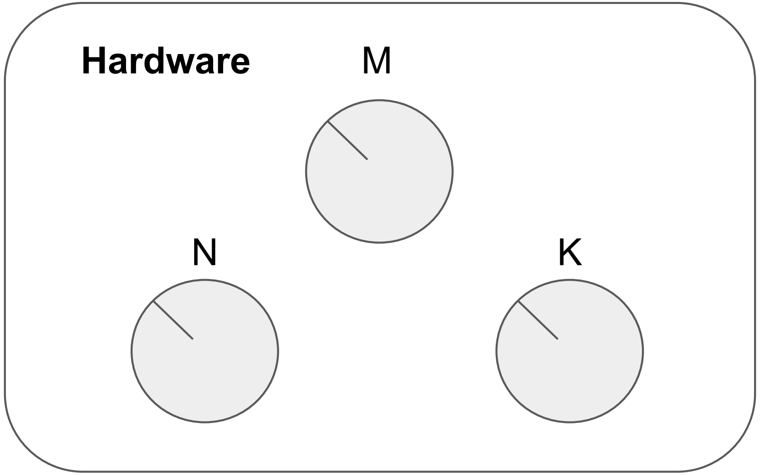
- HyperNetwork
  - Main Net
    - Linear
    - GRU Cell
    - FIR Filter: Latency Recognition
  - Hyper Net: MLP





Dataset

- Dataset
  - training: 6.5 min
  - valiation: 1.5 min
- AFx
  - Analog
    - Amp Distortion (mono-mono)
    - Sound Image Modifier (stereo-stereo)
    - Saturator (mono-mono)
  - Digital
    - phaser/flanger
- Sampling Rate
  - 48x2
  - 96x2



Combinations = N x M x K x ….

Loss

- Goal
  - Waveforms are **identical**
  - **phase** is considered
- STFT (Multi-Scale) **Complex** Spectrogram
  - similiar with ICASSP'21 paper from meta

Loss Function. To train our model, we minimize the multi-scale Short-Time Fourier Transform (STFT) loss [27], which has been commonly used to replace point-wise losses on the raw waveforms. Let L_i define a single STFT complex spectrogram l_1 loss with a given FFT size i. The total loss is then the sum of all the spectral losses for the left and right channels $L_{\text{total}} = \sum_i L_i^{(\text{left})} + \sum_i L_i^{(\text{right})}$. We use FFT sizes (2048, 1024, 512, 256), and the neighboring frames in the STFT overlap by 75%.

Baselines

- Model
  - WaveNet [1, 2, 3] [Concatenation]
  - RNN [3] [Concatenation]
  - micro-TCN [4] [FiLM]
  - hyper-conditioned IIR [5] [HyperNetworks]
  - hyperGRU (**proposed**) [HyperNetworks]
- Loss
  - Temporal domain losses [1, 2, 3, 4]
  - STFT-magnitude [6]
  - Hybrid [4]
  - STFT-complex (**proposed**) [7]

Baselines

[1] (SMC’19) Real-Time Modeling of Audio Distortion Circuits with Deep Learning

[2] (Icassp’19) Deep Learning for Tube Amplifier Emulation

[3] (Dafx’19) Real-Time Black-Box Modelling With Recurrent Neural Networks

[4] (Aes’22) Efficient Neural Networks for Real-Time Modeling of Analog Dynamic Range Compression

[5] (Icassp’21) Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads

[6] (ICLR’20) DDSP

[7] (ICLR’21) Neural Synthesis of Binaural Speech from Mono Audio

Experiments

- Oberservation 1: RNN > TCN
  - Quality
  - Model size
  - Efficiency (on CPU, Eigen C++)

Source. Run on Eigen C++

Table 2: *Error-to-signal ratio and processing speed for the Wavenet and proposed LSTM models of the Big Muff pedal. The best results are highlighted.*

Model	Hidden Size	Layers	Number of Parameters	ESR	Time (s) / s of Output
WaveNet	16	10	24065	11%	0.53
WaveNet	8	18	11265	9.9%	0.63
WaveNet	16	18	43265	9.2%	0.91
LSTM	32	1	4513	10%	**0.12**
LSTM	48	1	9841	6.1%	0.18
LSTM	64	1	17217	**4.1%**	0.24

Source. Run on Libtorch

Model	K	N	d	C	P	**R.f.**	**RT (CPU/GPU)**	**MAE ↓**	**STFT ↓**	**LUFS ↓**
TCN-324-N [20]	15	10	2	32	162 k	324 ms	0.5x / 17.1x	1.70e-2	0.587	0.520
TCN-100-N	5	4	10	32	26 k	101 ms	4.2x / 37.1x	1.58e-2	0.768	1.155
TCN-300-N	13	4	10	32	51 k	302 ms	1.8x / 37.3x	**7.66e-3**	0.600	0.602
TCN-1000-N	5	5	10	32	33 k	1008 ms	0.5x / 26.4x	1.20e-1	0.736	0.934
TCN-100-C	5	4	10	32	26 k	101 ms	5.0x / 37.2x	1.92e-2	0.770	1.225
TCN-300-C	13	4	10	32	51 k	302 ms	2.2x / 37.3x	1.44e-2	0.603	0.761
TCN-1000-C	5	5	10	32	33 k	1008 ms	0.6x / 26.4x	1.17e-1	0.692	0.899
LSTM-32	-	-	-	-	5 k	-	0.9x / 2.8x	1.10e-1	**0.551**	**0.361**

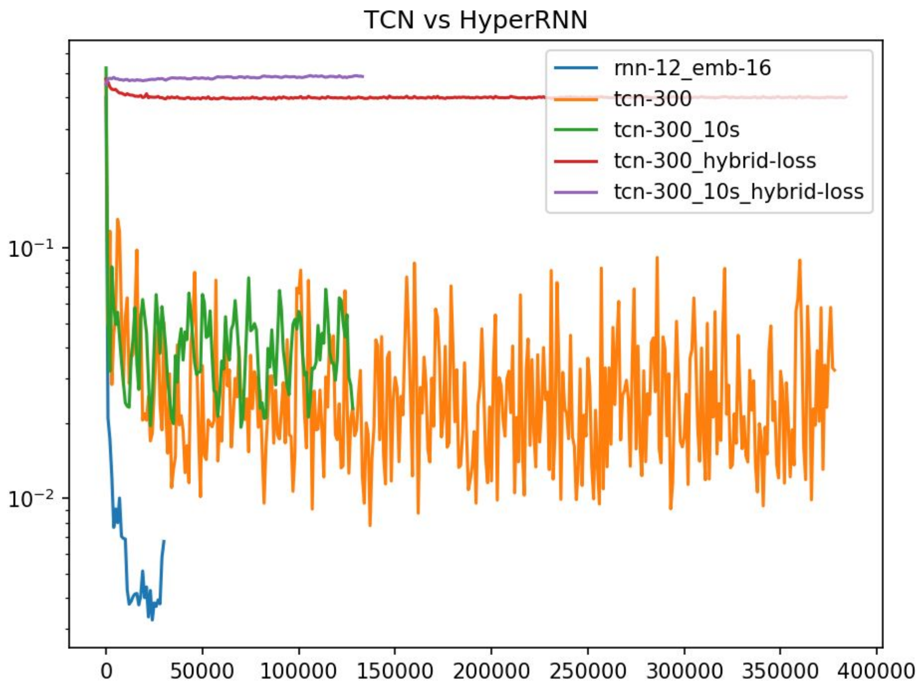
Experiments

- Oberservation 1: RNN > TCN
- The Efficiency largely depends on the platform and C++ framework
  - To achieve similar quality:
  - parameters amount: TCN >> RNN
  - speed on GPU: TCN > RNN
  - speed on CPU: (different framework)
    - libtorch   TCN > RNN
    - Eigen   TCN < RNN

- In deployment, we care **quality**, **model size** and **speed** on **CPU**
  - RNN > TCN

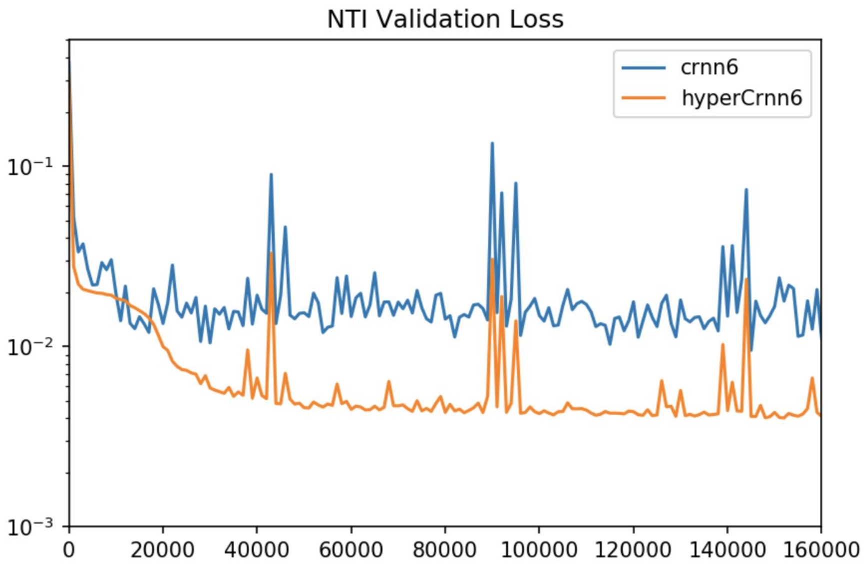
Experiments

- Oberservation 1: RNN > TCN
  - On our dataset



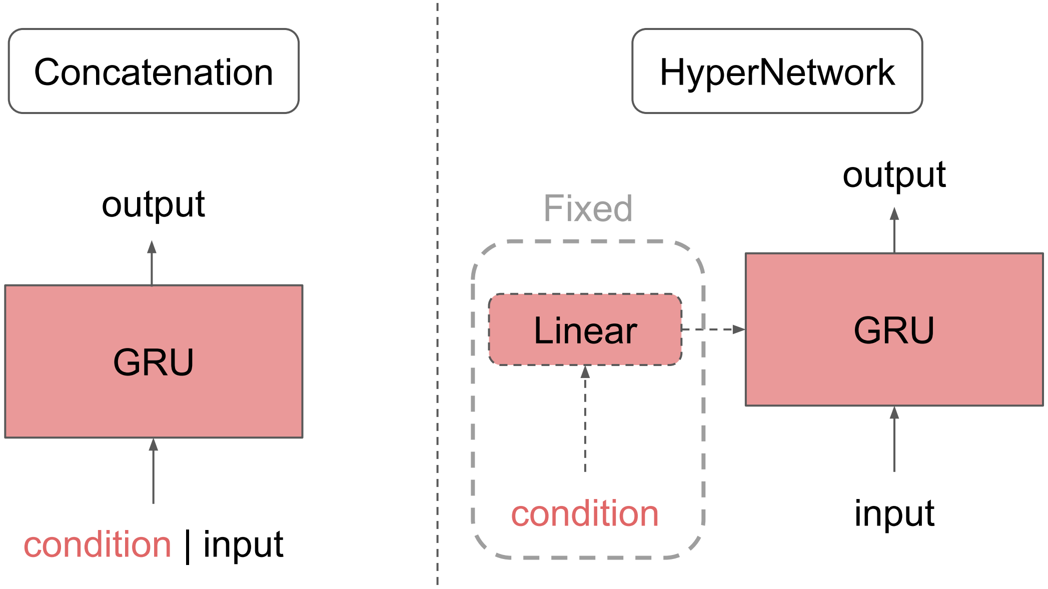
Experiments

- Oberservation 2: HyperGRU > Concatenation GRU
  - Quality



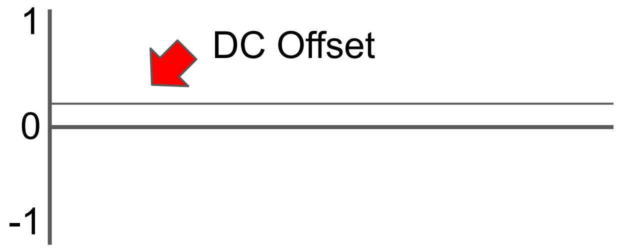
Experiments

- Oberservation 2: HyperGRU > Concatenation GRU
  - Efficiency

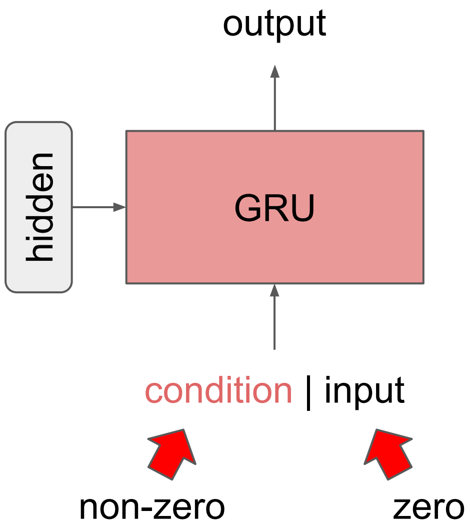


Experiments: DC Bias

- Reason
  - silence (zero) input
  - Concatenation GRU
  - Condition is **non-zero**



- Cold Start Issue
  - The steady hidden state of RNN is variable
  - Pop sound when open the plugin



Experiments: DC Bias

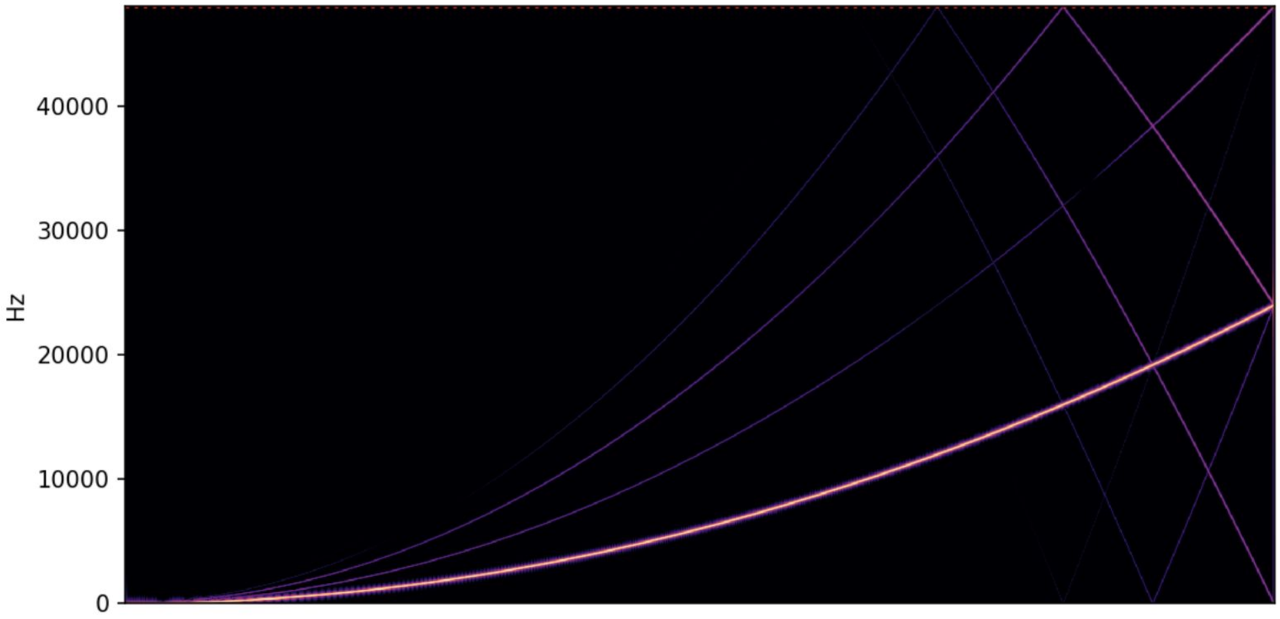
- (ICLR'21) [Neural Synthesis of Binaural Speech from Mono Audio]

Inspired by the DSP formulation, we predict the convolutional weights for the input $\boldsymbol{x}_{1:T}$ of a layer and the bias as functions of the conditioning input $\boldsymbol{c}_{1:T}$,

$$\boldsymbol{z}_t = \sum_{k=1}^{K} \left[\mathcal{H}^{(\mathbf{W})}(\boldsymbol{c}_{1:t})\right]_{:,:,k} \boldsymbol{x}_{t-k+1} + \mathcal{H}^{(\boldsymbol{b})}(\boldsymbol{c}_{1:t}). \tag{6}$$

- Model Design
  - HyperNetworks
  - Model Bias = False

Experiments: Aliasing



Even self reconstruction has this problem: tanh, sigmoid

Experiments: Aliasing

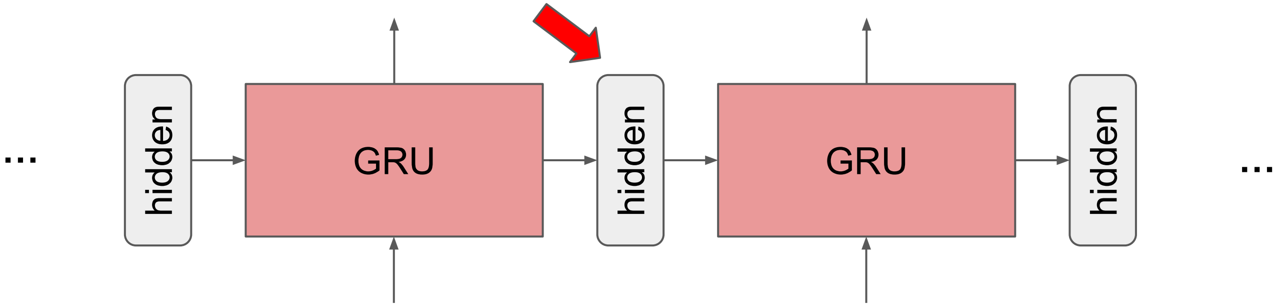
- Solution: Oversampling
- (Interspeech’20) Real Time Speech Enhancement in the Waveform Domain

Finally, we noticed that upsampling the audio by a factor U before feeding it to the encoder improves accuracy.

Experiments: Training

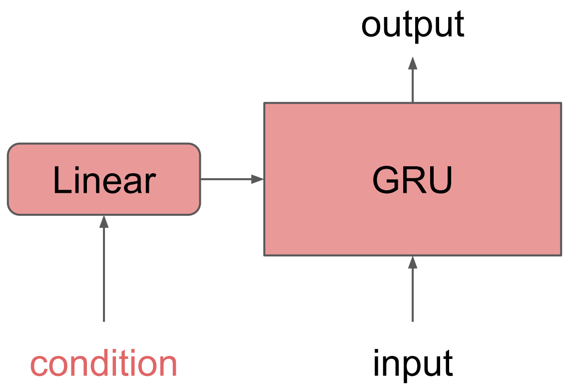
- Truncated BPTT
  - Buffer by buffer
- Passing Hidden State Across Buffer
  - faster convergence
  - higher quality

Policy of hidden state:
- Zero or random Initialization
- **Pass**



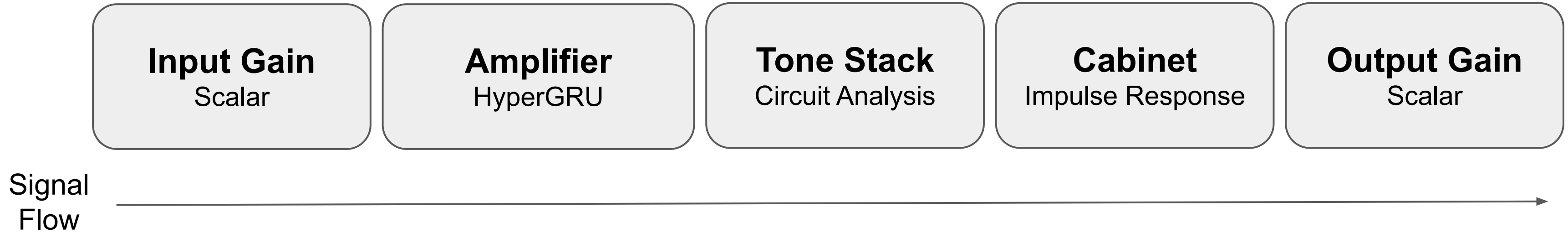
Deployment

- C++ Framework
  - JUCE
  - Eigen C++
- HyperNetwork Update Policy
  - No chagne in condition: fixed
  - Changed
    - interpolation
- RTF = 0.2 - 0.3 (stereo) on CPU



Deployment

- Difficulty in Dataset Construction
  - Combinations
- Hybrid Method

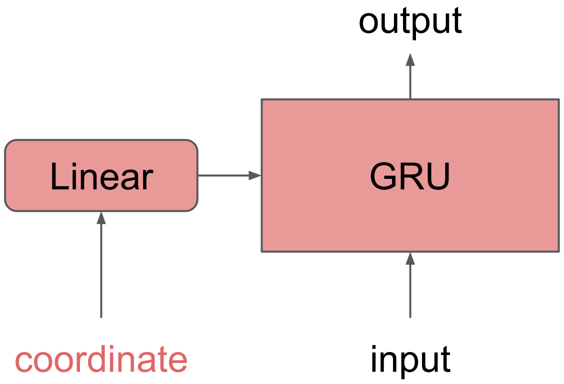


Chapter 2 - HyperGRU for Neural AFx Modeling

- Current Progress
    - Model
    - Dataset
    - Loss
    - Baselines
    - Experiments
    - Deployment
- **The Palette**
    - **Tone Creation**
    - **Discussion**
- Future Work
    - Advanced Model Design
    - Benchmark
    - Discussion

Tone Creation

- Tone Creation / Fushion
- Crate an embedding for tones



Tone Creation

- Inspired by these works
  - (arXiv.2010) Randomized Overdrive Neural Networks
  - (NeurIPS'21) Steerable Discovery of Neural Audio Effects (ML4CD Workshop)

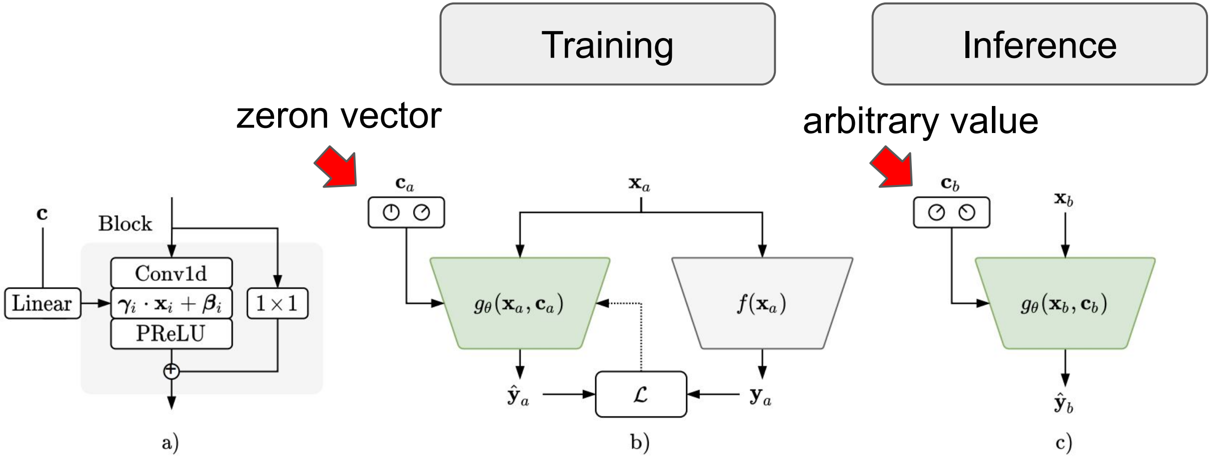


Figure 1: a) TCN block with 1D convolution, conditional affine transformation (FiLM), followed by a PReLU nonlinearity. b) Steering process where $g_\theta(\mathbf{x}_a, \mathbf{c}_a)$, a conditional TCN, is trained to emulate $f(\mathbf{x}_a)$, an existing audio effect, using a single input/output pair of recordings $\mathbf{x}_a, \mathbf{y}_a$. c) Generation process where $\mathbf{x}_b$, a new signal, is processed with the TCN and differing conditioning parameters $\mathbf{c}_b$.

Tone Creation

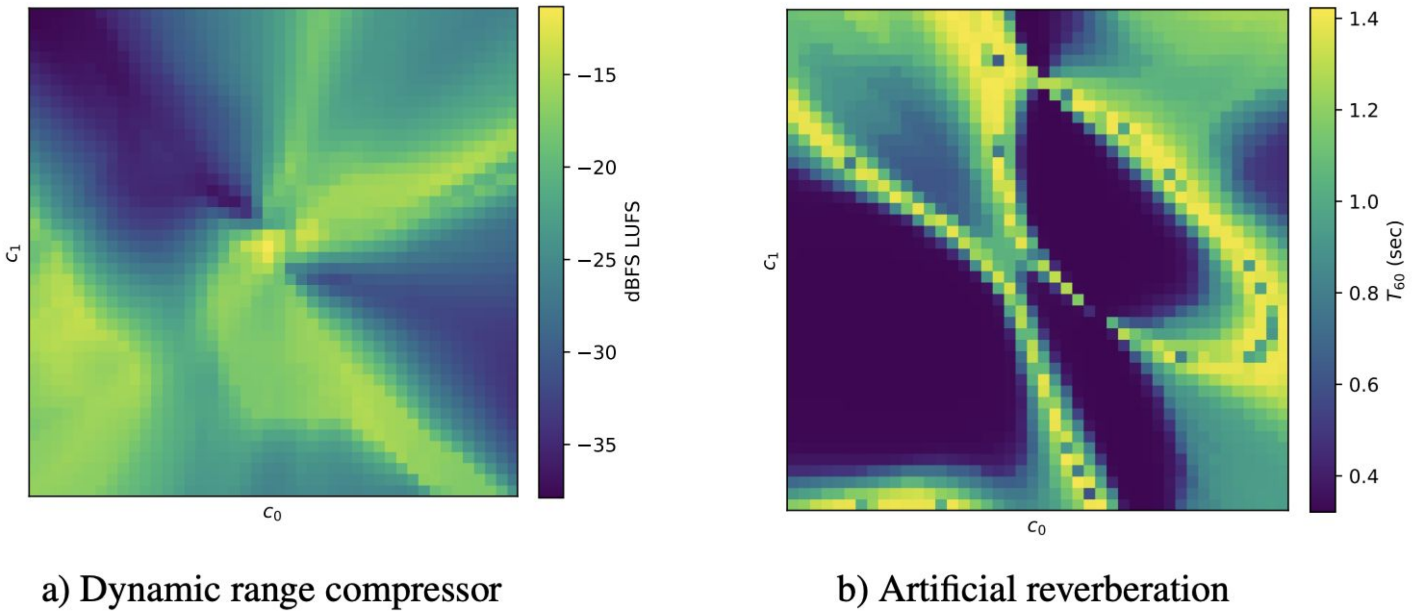
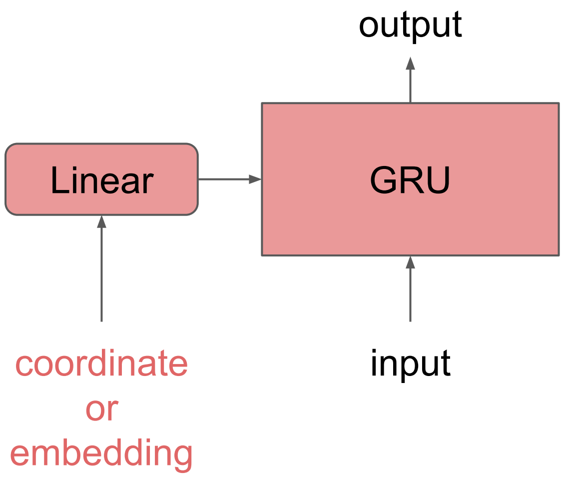


Figure 2: Parameter space $\mathbf{c} \in \mathbb{R}^2$ from -5 to 5 with relation to a) loudness dB LUFS for a model steered with a signal from a dynamic range compressor, and b) T_{60} for a model steered with a signal from an artificial reverberation effect, both of which demonstrate clear structure.

Discussion

- VAE-like embedding might not be necessary
    - no distrubution?
    - 2D plane
        - interpolation
        - extrapolation
- Tone Creation
    - more tones
    - embedding projection
    - GAN

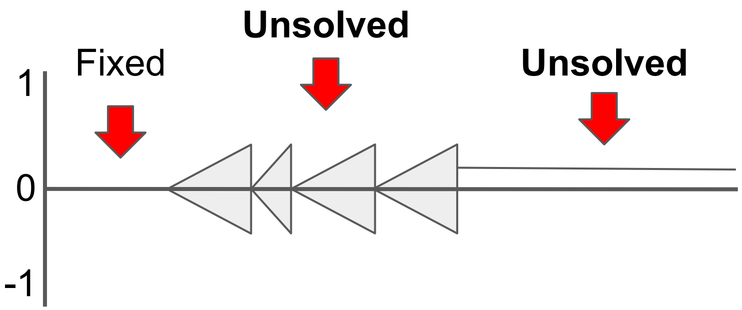


Chapter 2 - HyperGRU for Neural AFx Modeling

- Current Progress
  - Model
  - Dataset
  - Loss
  - Baselines
  - Experiments
  - Deployment
- The Palette
  - Tone Creation
  - Discussion
- **Future Work**
  - **Advanced Model Design**
  - **Benchmark**
  - **Discussion**

Advanced Model Design

- DC Bias
  - **post-silence**
  - runtime
- Aliasing



input: sine wave@1k

Advanced Model Design

- Possible Reason:
  - (ICLR'17) A Recurrent Neural Network without Chaos
- Solution?
  - RNN-Decay

Source:
(NeurIPS'19) Latent ODEs for Irregularly-Sampled Time Series

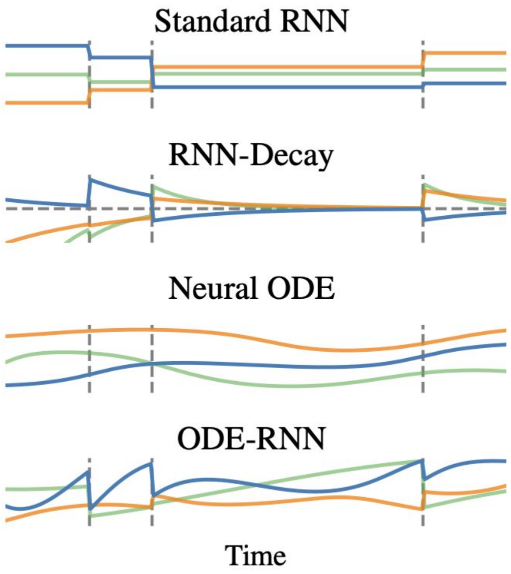


Figure 1: Hidden state trajectories. Vertical lines show observation times. Lines show different dimensions of the hidden state. Standard RNNs have constant or undefined hidden states between observations. The RNN-Decay model has states which exponentially decay towards zero, and are updated at observations. States of Neural ODE follow a complex trajectory but are determined by the initial state. The ODE-RNN model has states which obey an ODE between observations, and are also updated at observations.

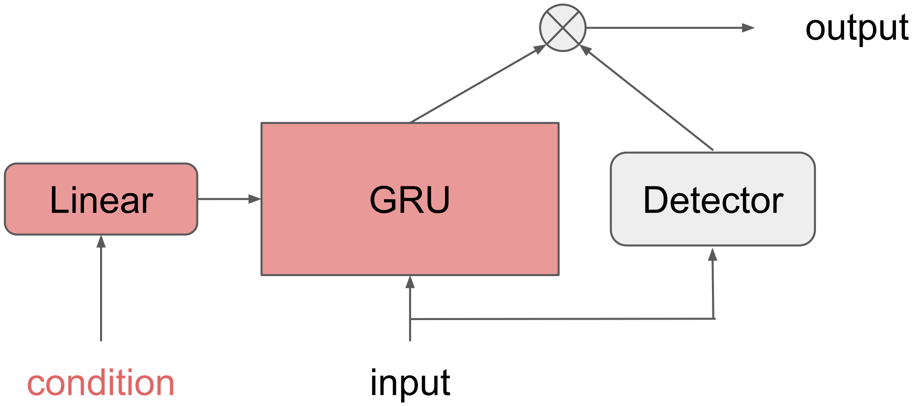
Advanced Model Design

- RNN-Decay
  - (NeurIPS'19) Latent ODEs for Irregularly-Sampled Time Series

observations are made [Che et al., 2018, Cao et al., 2018, Rajkomar et al., 2018, Mozer et al., 2017]:

$$h_i = \mathrm{RNNCell}(h_{i-1} \cdot \exp\{-\tau\Delta_t\}, x_i) \quad (2)$$

where τ is a decay rate parameter. However, Mozer et al. [2017] found that empirically, exponential-



Advanced Model Design

- Transient Modeling
  - (ISMIR'19) Deep Unsupervised Drum Transcription
  - Onset-enhanced loss

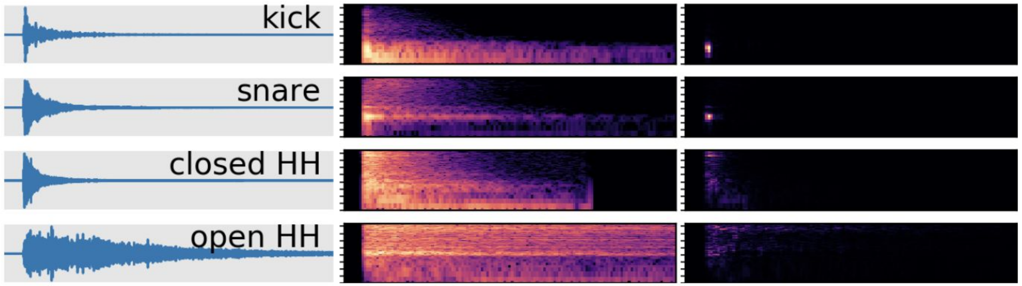


Figure 3: The effect of drum extraction for kick, snare, close hi-hat, and open hi-hat, from top to bottom. Columns are from left to right: original waveform, original spectrum, and onset-enhanced spectrum

Benchmark

- Dataset
- {TCN, RNN, IIR} x {Concatenation, FiLM, HyperNework}
- Integrated with DDSP components
- Losses

Discussion

- Sampling Rate Agnostic
  - Input of HyperNet is sampling rate

- HyperNet:
  - MLP/CNN/RNN?
  - Doppler Effect?
  - Few/zero shot learning?
  - ADAA?

- Chapter 4: Future Work

03

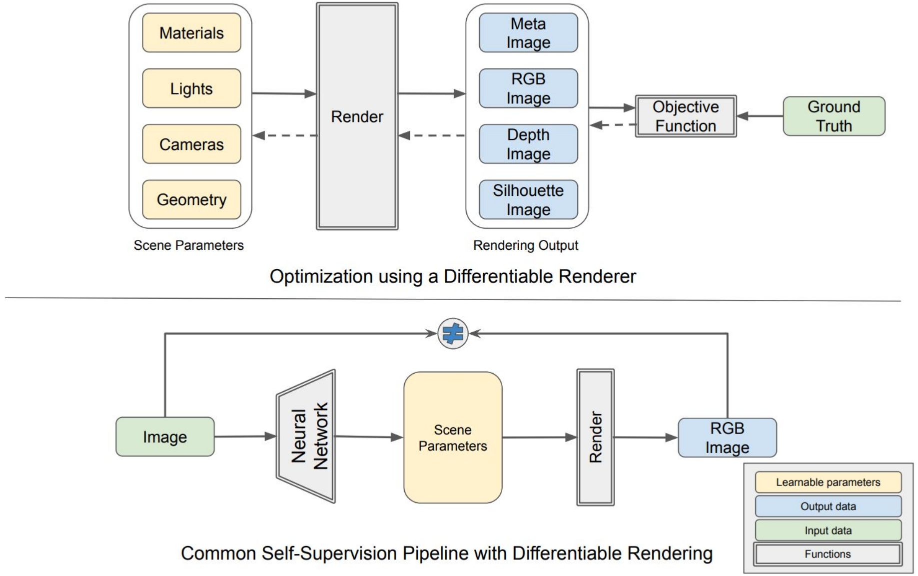
Future Work

Chapter III

Chapter 3 - Future Work

- **Technology**
  - **Intelligent Music Production**
  - **Digitization**
  - **Sound Field Reconstruction**
- Future of Creation

Intelligent Music Production



Near Future

- Channel Strip
- AI Guitar Tone

Vision

- AI Mixing/Mastering/Creation
    - Similiar Concept in Computer Vision: [Differentiable Rendering]() (arxiv.2006)

Channel Strip

- Coloring
  - product: The Cat
  - product: The Palette
  - product: British Kolorizer
- EQ
  - prototype: maag
  - prototype: Flickenger
- Dynamic
  - None (research required)

- AI Channel Strip
  - every part is differentiable

AI Guitar Tone

- Pedal
  - prototype: DS1
  - prototype: Digital Phaser/Flanger
- Amplifier
  - product: British Kolorizer
  - prototype: 5150
- Cabinet
  - IR

- AI Guitar Tone
  - every part is differentiable
  - amp/pedal palette

Digitization

Key: Sampling Rate Agnostic / Runtime Sampling

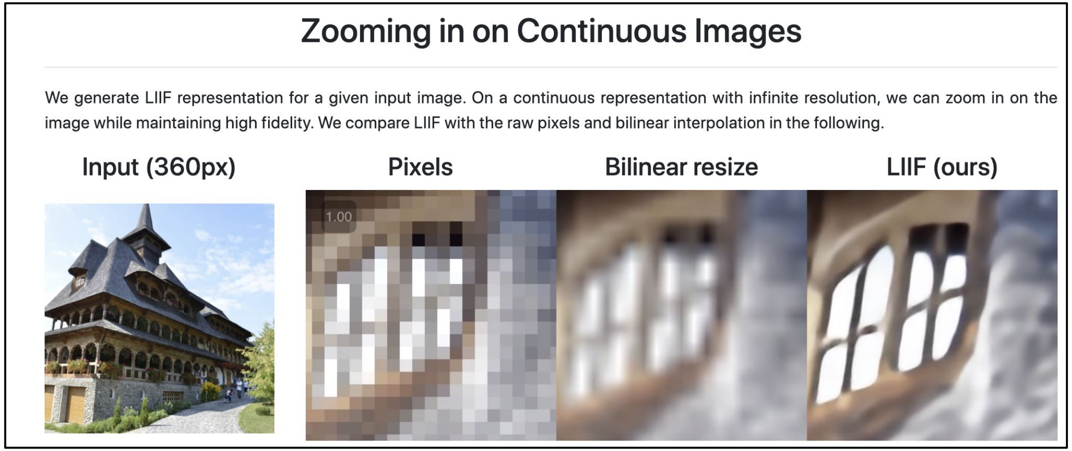
- Implicit Neural Represenation
- Continuous Domain Deep Learning

Implicit Neural Represenation

- (NeurIPS’20) SIREN: Implicit Neural Representations with Periodic Activation Functions
- (ECCV’20) NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
- (CVPR’21) Learning Continuous Image Representation with Local Implicit Image Function

Original: Image[x, y] = [R, G, B]

Implicit Neural Reprentation: Image(x, y) = [R, G, B]



Implicit Neural Represenation

- Computer Vision: Applications
  - Super Resolution
  - Novel View Synthesis

- Audio?
  - Sound Field Reconstruction

Continuous Domain Deep Learning

- CNN
  - (ICLR'22) CKConv: Continuous Kernel Convolution For Sequential Data
- RNN / Neural ODE
  - Uneven sampled time series: ΔT
  - (Dafx'22) Virtual Analog Modeling of Distortion Circuits Using Neural Ordinary Differential Equations
  - …

Sound Field Reconstruction

- Zhenyu Tang
    - (Interspeech'21) IR-GAN: Room impulse response generator for far-field speech recognition
    - (IEEE VR'21) Learning Acoustic Scattering Fields for Dynamic Interactive Sound Propagation
    - (arXiv.2204) **GWA: A Large High-Quality Acoustic Dataset for Audio Processing**

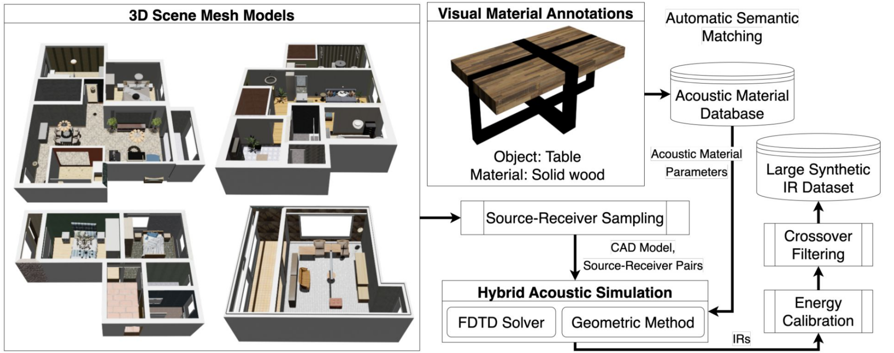
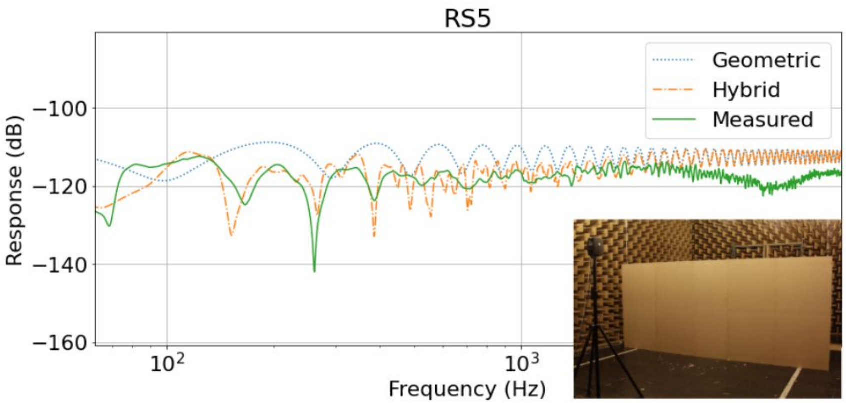


Figure 1: Our IR data generation pipeline starts from a 3D model of a complex scene and its visual material annotations (u
structured texts). We sample multiple collision-free source and receiver locations in the scene. We use a novel scheme to aut
matically assign acoustic material parameters by semantic matching from a large acoustic database. Our hybrid acoustic sir
ulator generates accurate impulse responses (IRs), which become part of the large synthetic IR dataset after post-processin;

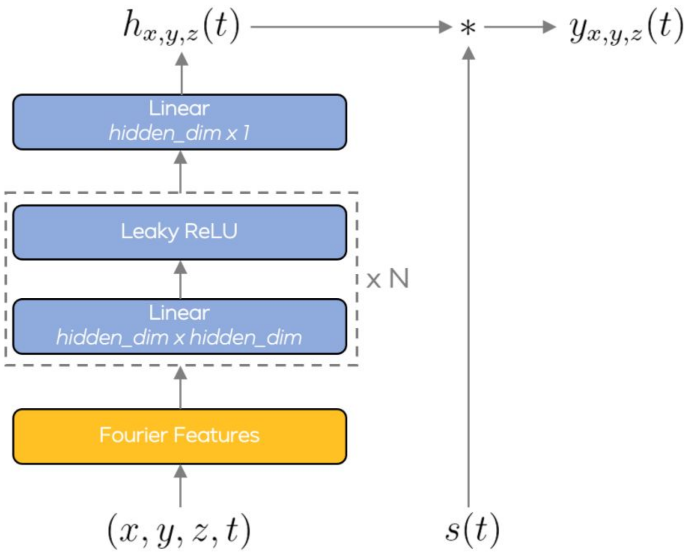


(a) RS5: simple diffraction with infinite edge.

source: 3D-FRONT Dataset

Sound Field Reconstruction

(arXiv.2202) [Deep Impulse Responses: Estimating and Parameterizing Filters with Deep Networks]
(arXiv.2204) [Learning Neural Acoustic Fields]



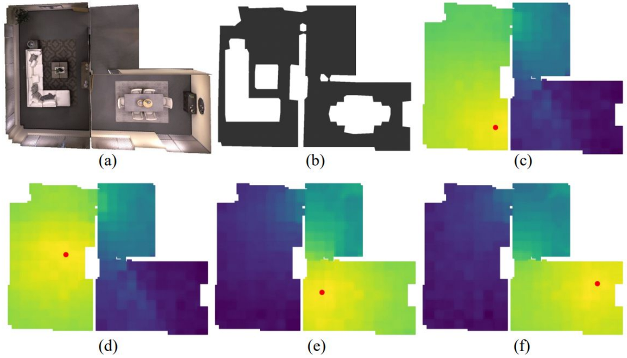
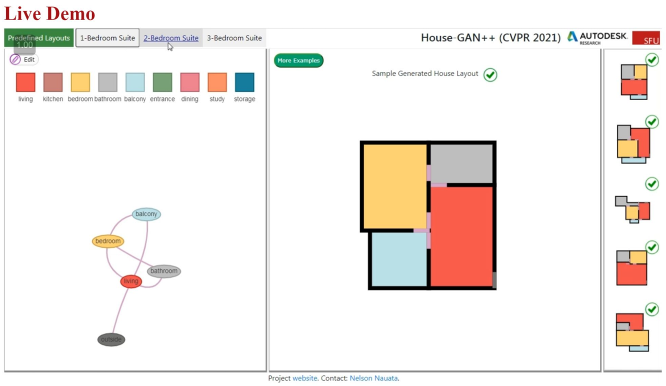


Figure 1. Neural Acoustic Field (NAF) learns an implicit representation for acoustic propagation. **(a)** A 3D top-down view of the house with two rooms. **(b)** Floor of the rooms shown in grey. **(c)-(f)** The loudness of acoustic field as predicted by our NAF is visualized for an emitter located at the red dot. Notice how sound does not leak through walls, and the portaling effect open doorways can have. Louder regions are shown in yellow.

Sound Field Reconstruction

- Given a 3D object (indoor scene), recreate the sound field
  - reverb plugin
    - best in the industry: altiverb
  - wayverb
- What if the 3D model is also generated by AI
  - (CVPR'21) House-GAN++: Generative Adversarial Layout Refinement Networks



Future of Creation

- Observation
  - From 2D to 3D
  - From Digital to Analog
  - High Quality
  - Focus on “Concepts”
  - Immersive Experience
    - Dolby ATMOS
    - Ambisonic
  - Knowing, then can creation

https://www.blender.org/

Future of Creation

- POC
  - Virtual room
  - Genre
  - Sound field
  - Music
    - AFx
    - materials
  - Interactive web

Thank you